

UNIVERSIDAD COMPLUTENSE DE MADRID

FACULTAD DE INFORMÁTICA

Departamento de Ingeniería del Software e Inteligencia Artificial



TESIS DOCTORAL

Planificación y supervisión de flotas de robots autónomos en tareas agrícolas

MEMORIA PARA OPTAR AL GRADO DE DOCTOR

PRESENTADA POR

Jesús M^a Conesa Muñoz

Directores

**Ángela Ribeiro Seijas
Gonzalo Pajares Martinsanz**

Madrid, 2016

UNIVERSIDAD COMPLUTENSE DE MADRID
FACULTAD DE INFORMÁTICA

Departamento de Ingeniería del Software e Inteligencia Artificial



TESIS DOCTORAL

PLANIFICACIÓN Y
SUPERVISIÓN DE FLOTAS DE
ROBOTS AUTÓNOMOS EN
TAREAS AGRÍCOLAS

Jesús M^a Conesa Muñoz

Madrid, 2015

UNIVERSIDAD COMPLUTENSE DE MADRID
FACULTAD DE INFORMÁTICA

Departamento de Ingeniería del Software e Inteligencia Artificial



PLANIFICACIÓN Y SUPERVISIÓN DE FLOTAS DE ROBOTS AUTÓNOMOS EN TAREAS AGRÍCOLAS

MEMORIA PARA OPTAR AL GRADO DE DOCTOR
PRESENTADA POR

Jesús M^a Conesa Muñoz

Bajo la dirección de los doctores

Ángela Ribeiro Seijas
Gonzalo Pajares Martinsanz

Madrid, 2015

Departamento de Ingeniería del Software e Inteligencia Artificial
Facultad de Informática
Universidad Complutense de Madrid

Tesis Doctoral

Planificación y supervisión de flotas de robots autónomos en tareas agrícolas

Autor

Jesús M^a Conesa Muñoz

Directores

Dra. Ángela Ribeiro Seijas

Dr. Gonzalo Pajares Martinsanz

ISBN 978-84-608-3371-0
Madrid, Noviembre 2015

Si se cree y se trabaja, se puede
(Diego Pablo Simeone)

La investigación presentada en este trabajo ha sido financiada por el 7º Programa Marco de la Comunidad Europea [FP7/2007-2013] bajo el acuerdo de subvención nº245986.

Agradecimientos

Nunca estuvo entre mis planes escribir una tesis doctoral, ni siquiera sabía realmente en qué consistía, solo cuando tuve la suerte de empezar a trabajar en el antiguo Instituto de Automática Industrial (hoy Centro de Automática y Robótica), empecé a hacerme una idea. En esos comienzos, hace ya 7 años, veía a gente que iba y venía, todos ellos con una misma obsesión, *la tesis...* Para mí era un tema bastante ajeno en el que apenas me paraba a pensar. No es que no quisiera hacer un doctorado, es que no me lo planteaba, entre otras cosas porque ni sabía el tiempo que estaría allí. Sabía que la vida da muchas vueltas y que lo que piensas hoy puede ser muy diferente a lo que hagas mañana. ¿Cómo entonces he acabado escribiendo esto? Pues por la Dra. Ángela Ribeiro, ella fue quien confió en mí y quien poco a poco fue orientándome para que me lo propusiera, la mayoría de las veces incluso con más convicción que la mía propia. Así que no dudo ni un segundo en reconocer que esta tesis nunca se hubiera escrito sin ella. Este trabajo es en muy gran medida el suyo. Gracias, Ángela.

Los otros grandes responsables son, como no podía ser de otro modo, mi familia. Sin ellos sería una persona totalmente diferente y con toda seguridad para peor, incluso más protestón. Ellos me han enseñado cuál es el camino, y no solo con palabras, sino también, y sobre todo, con ejemplo y amor. A ellos va dedicada esta tesis. Especialmente a mi padre y mi abuelo, quienes están conmigo a diario, y a los que ojalá algún día pueda parecerme. Gracias, papá, mamá, abuelos, Isabel, Álvaro, Jaime, Laura, Ramón, Susana, Loreto, y a todos los que estén por llegar.

Quiero también agradecer su apoyo al Dr. Gonzalo Pajares, por su gran disposición y por su siempre útil dirección, así como al resto de Complutenses con los que he tenido el placer de trabajar. Especialmente a Martín y José Miguel, dos señores con los que uno puede acabar perdiéndose en la Lérida más profunda y disfrutar de ello.

No puedo dejar de agradecer al Dr. Pablo González de Santos darme la oportunidad de trabajar en RHEA, proyecto que ha dado lugar a ésta y a tantas otras tesis, y a los doctores José Dorado y César Fernández-Quintanilla por compartir su gran experiencia agrícola conmigo.

Gracias también a todos los compañeros de RHEA y muy especialmente a Mariano, Luis y Joao, por su ayuda en todas esas horas de pruebas y tensión que pasamos juntos.

Por supuesto, también quiero dar las gracias a todos mis compañeros del Centro de Automática y Robótica por aguantarme durante todos estos años. Realmente tiene mucho mérito. Voy a tratar de acordarme de todos, aunque probablemente me deje a alguno sin querer. Si es así, le pido que me perdone.

A Almudena y Domingo, mis primeros compis de despacho y personas de bondad infinita que me integraron rápidamente y me hicieron sentir desde el primer día como otro (bicho raro) más del centro.

A David Martín, próximo premio Nobel de la paz, por su amabilidad y disposición, y por acompañarme en aquellos Robocity sorpresa que surgían espontáneamente de la nada.

A Xavi, Nacho y Patri, por todas esas quedadas, horas de Quake y bromas anónimas aún sin resolver...

A Esther y e-R1k4, de Oruga, por aguantar mis chistes machistas sobre vuestro peso, color de piel, acento, nivel de inglés, color de labios, coche, conocimientos de deporte... Vosotras sois de las que más os lo habéis ganado. ¡Vivan las d...!

A José Pedro, la avutarda de Triad, por ser la única persona y/o pájaro normal de todo el centro, y la única persona de toda España capaz de ser objetiva en temas de fútbol o política... Sí, algo raro tiene que esconder, y a juzgar por su volumen...

A Nadir y Javi Herrera, por aguantar mis presentaciones de varias horas y mis mono-temas de conversación.

A Manuel, el *homo gorrinensis*, por nuestras tertulias profundas sobre historia clásica, literatura francesa, filosofía contemporánea, alta teología, etc. Y por ser un tío sencillo y noble. Cuando sea mayor quiero ser como tú, Manuel, pero cuando sea muy mayor.

A Ranz, Gilí Ranz... tú sí que has aguantado, pero también eres de los que más cuerpo tiene para ello... por tu ingenio y buen humor, por lo que luchamos en Vietnam y en Endora, por estar siempre ahí, literalmente (no te ibas de mi despacho ni con agua caliente), cuando hacía falta y sobre todo cuando no.

A Raquel Sáez y Juan Carlos, por su buen humor en los desayunos y quedadas.

A todas las Carmen y Mari Carmen, a Rosi y a Petri, por todas esas tardes haciéndome compañía.

A Andrea, por ser mi colombiana favorita. And to Andrew, because you are my favourite Irish person. Sois un poco mejores que los otros dos que conozco de esos países.

A Chema, por buena persona, por soportar todas mis quejas, por escucharme siempre, por ayudarme en todo y por ser un madridista de pro, de los que hacen grande a su equipo.

A Antonio y Lisa, por las tertulias tan amenas en el despacho. ¡Y a Isabel! La niña prodigio que me va a quitar el trabajo dentro de un par de años.

A Olp, otro madridista buena gente. Por ser tan buen tío y por caerte bien desde el primer momento sin ninguna explicación posible. Eeeeeese Oooooo!p.

A Óscar weeeey, por su buen humor, por enseñarme a hablar bien chido. Eso estuvo padre.

A Dioni y Andrés, los nuevos. A Andrés por tener la deferencia de esperar a que acabe completamente la tesis parar plagiarle, y a Dioni por ser del Atleti, con eso ya has hecho más que casi todos los demás.

A Dalmay, por tu buen corazón y por no mandarme a freír espárragos cuando te aseguraba que Cuba estaba en Sudamérica o cuando te preguntaba si sabías hacer arroz a la cubana.

A Giovanna, Vander y Marcelo, por sua simpatia e alegria brasileira.

A Cruza, por ser un rival digno en el Quake, por no dejarme sordo, por no caminar desnudo por el pasillo, por entender mis chistes sobre Futurama, tetraedros o la vanguardista forma de pensar de Manuel... En definitiva, por todo lo importante.

A Marga, por su simpatía y cercanía, y por obligarme a ir a desayunar de vez en cuando.

A Luis, el argentino, mi compi de mesa en el comedor. Por las tertulias tan agradables, por las recomendaciones cinéfilas y por explicarme las diferencias entre un cumulonimbo y un cirrostrato. ¡Ahora ya no los confundo cuando me los encuentro!

A Carlos Julián, por su muy agudo humor y por ser la única persona capaz de rivalizar conmigo en quejas, protestas, enfurruñamientos... Sin duda ese súper poder te dio la capacidad para entenderme.

A Sofía, o Sonia, o Mari Sofi, o José Fina, por su simpatía, inocencia y alegría.

A David Romero y Javi Villazón, por perdonarme por no dedicaros un capítulo entero de agradecimientos. Porque de verdad que no creo que vaya a tener compañeros mejores que vosotros. *Agradeciqui, changuex* y ¡larga vida al tirano!

A Fluis, por no pegarme aquel día que alguien te llamó por teléfono... por soportar todos mis comentarios constructivos sobre tu campo de energía Fatal, mis observaciones positivas sobre tu Forma Física, mi respetuosa e informada opinión sobre el bable... ¡Te has ganado el cielo, Fluis!

A José Miguel y Josito, por las tertulias futboleras.

A Jose, el informático, porque seguramente haya evitado varias caídas mundiales de internet y la explosión de alguno de los experimentos de José Pedro.

A Nuria, por ser la única persona capaz de darse cuenta de que no me inventaba el 90 % de las cosas, sino más bien el 100 %, ¡y discútmelo y obligarme a reconocerlo!

A Ángel, mi socio, porque podríamos estar un millón de años sin vernos y aún así nos entenderíamos con un simple gesto. Porque somos del *equipo campeón, los mejores porque sí*.

A Raquel Galán, mi otra socia, por ser encantadora y la más guapa del centro.

A Damià, el tío més honrat, més sensat y amb més germans que conec. Perquè els dos sabem que aquesta tesi s'ha escrit *malgrat* tots els altres.

Y, por supuesto, también quiero agradecerse a muchos otros más allá del ámbito laboral, porque también han participado de un modo u otro.

A Rubén, porque si no hubiera sido por él nunca hubiera mandado mi CV al CSIC. Y a Josa, porque es el que siempre va con Rubén y por enseñar al mundo que si no tienes tu propio grosellero estás perdiendo dinero, y *lo sabes*. ¡10 para mí!

A Sirlei, Valmir, Danieli y Natalia, muito obrigado pelo vosso apoio.

A Cardoomx, por sus sabios consejos.

A Guillermo, Dani, Jon y Chavo, por tantos momentos de diversión.

A Alex, por ser mi amigo, por ver siempre lo bueno que hay en mí sin prestar nunca atención a lo malo.

A Javi y David, por compartir risas e historias míticas, y por formar parte del mejor grupo de whatsapp de la historia.

A Raquel, por su sonrisa radiante que todo lo ilumina.

A Almudena, por ser perfecta.

Y a Dios, que en su infinita bondad y misericordia me ha dado todo lo que necesito.

¡GRACIAS!

Resumen

El control seguro y efectivo de las malas hierbas según los principios establecidos por la Agricultura de Precisión requiere una tecnología específica, avanzada y de coste asumible dentro de los márgenes de beneficio. El uso de plataformas móviles autónomas tanto terrestres como aéreas equipadas con sistemas de percepción innovadores, sistemas inteligentes de toma de decisión y herramientas que permitan la aplicación precisa de los tratamientos herbicidas, reduce el coste asociado a la operación así como los potenciales daños ambientales y el riesgo para los agricultores. Varios son los grupos internacionales de investigación dedicados al desarrollo de tecnologías basadas en sistemas robóticos capaces de optimizar las operaciones complejas implicadas en este tipo de tratamientos de precisión.

El desarrollo de sistemas autónomos de ayuda al tratamiento preciso, tanto de cultivos extensivos (malas hierbas) como de leñosos, es el objetivo principal del proyecto RHEA (Robot Fleets for Highly Effective Agriculture and Forestry Management), financiado por el 7º Programa Marco de la CE y dirigido a minimizar los insumos (agroquímicos, combustible, etc.) a la vez que garantizar la calidad y seguridad del producto así como cubrir totalmente el campo independientemente del tamaño que éste tenga. RHEA propone utilizar una flota de robots pequeños/medianos tanto aéreos como terrestres para la inspección/monitorización y posterior aplicación de fitosanitarios, lo que presenta múltiples ventajas frente al tratamiento basado en la habitual máquina de mayores dimensiones y más tradicional. Entre otras, una flota de robots de tamaño pequeño/mediano reduce el impacto sobre la compactación del suelo e interactúa de un modo más seguro con los operarios, ya que la detección e interacción se puede distribuir en varios sistemas de detección y gestión de fallos como los propuestos en la presente tesis.

El trabajo de investigación presentado en esta tesis se ha desarrollado dentro de este proyecto europeo y está relacionado con el diseño, desarrollo y evaluación del nivel más alto de la arquitectura RHEA, en otras palabras, con los tres aspectos fundamentales para conseguir que los robots de la flota ejecuten el trabajo eficientemente y sin intervención humana, es decir, con la planificación, la supervisión y la gestión completa e integrada de las tareas de inspección y tratamiento.

La planificación es necesaria para obtener las trayectorias que deben seguir las distintas unidades terrestres de la flota para llevar a cabo la tarea coordinadamente. En concreto, en la tesis la obtención del plan se plantea como un problema de optimización combinatoria cuyo objetivo es determinar las rutas y operaciones que optimizan la ejecución de la tarea en términos de algún criterio (coste por insumos, tiempo requerido, distancia recorrida, etc.). La tesis formula el problema de una forma muy general para abarcar multitud de escenarios agrícolas reales. A saber, usando uno o varios vehículos a la vez, tanto con iguales o distintas características, con limitaciones físicas reales, como radio de giro o capacidad de almacenamiento de productos, tanto de fitosanitarios como de combustible. También se ha considerado la posibilidad de hacer paradas de reabastecimiento y se ha tenido en cuenta la variabilidad del cultivo, por ejemplo, mediante mapas de distribución de rodales de malas hierbas. Además se ha trabajado con cualquier forma de campo e incluso con campos compuestos por varias parcelas. Por último, se han encontrado los mejores planes para cada vehículo de la flota considerando diferentes criterios, bien combinándolos adecuadamente en un único objetivo o mediante

optimizaciones multi-objetivo. El problema de optimización formulado se resuelve utilizando y comparando tres técnicas meta-heurísticas: Recocido Simulado, Algoritmos Genéticos y Non-dominated Sorting Genetic Algorithm II (NSGA-II). Además, para acelerar la convergencia de estos métodos, se ha propuesto y probado un operador, mix-opt, que construye nuevas soluciones a partir de la combinación ponderada de los operadores más utilizados en problemas similares.

Una vez encontrados los planes óptimos para los elementos de la flota, el siguiente paso consiste en ejecutar de forma autónoma los planes y comprobar que la ejecución se realiza según lo planeado. En los sistemas autónomos puede suceder que el comportamiento se desvíe de lo esperado por su propia complejidad y, sobre todo, por la incertidumbre de los elementos externos que les afectan, muy impredecibles y difícilmente controlables, especialmente cuando se trabaja en exteriores, como sucede en el marco de trabajo de esta tesis. Por tanto, es de vital importancia advertir las desviaciones del plan cuanto antes para poder reconducir la actuación de los sistemas o incluso detenerlos, sobre todo cuando los sistemas controlan maquinaria móvil de cierta envergadura y, por tanto, potencialmente peligrosa. En este contexto la tesis propone una arquitectura de supervisión distribuida y en varios niveles basada en los conceptos de supervisor y alarma. Estos dos conceptos permiten desacoplar la supervisión en módulos independientes y fáciles de implementar, que pueden estar distribuidos, y, puesto que los supervisores pueden emplearse para monitorizar otros supervisores, permiten construir incrementalmente un gran sistema global de supervisión en varias capas, desarrollando primero niveles bajos dedicados a la monitorización de los comportamientos más sencillos y, después, niveles más altos, dedicados al control de los comportamientos más complejos que son fruto de la combinación de los primeros. A partir de esta arquitectura se diseña un sistema de supervisión de varios niveles. El nivel más bajo opera dentro de las unidades terrestres y detecta sus problemas de forma individual. Un nivel más alto para supervisar aquellos aspectos que involucran a más de un elemento de la flota a la vez, por ejemplo, el posible riesgo de colisión entre dos o más vehículos. Y, por último, un tercer nivel que centraliza toda la información para que el operario responsable de la tarea monitorice la actuación de toda la flota y tome el control en caso de ser necesario.

Por último, la tesis propone un sistema Gestor para alcanzar la plena integración entre la monitorización/inspección llevada a cabo por las unidades aéreas de la flota y la intervención/tratamiento llevado a cabo por las unidades terrestres de la flota. El Gestor permite que un único operario defina los datos de entrada de las planificaciones y supervise la generación de los planes y la ejecución de los mismos en las unidades como parte de una misión, gestionando además la información de monitorización generada por la supervisión. Es decir, un sistema que enlace los diferentes pasos para llevar a cabo una tarea agrícola de tratamiento de forma totalmente automatizada. Por supuesto, unido a esta gestión completamente automática de la tarea/misión existe un sistema que informa en tiempo real del estado de todos los elementos de la flota que estén involucrados en la misión.

Los tres aspectos, planificación, supervisión y gestión, se han evaluado con numerosos ensayos teóricos y prácticos. Entre ellos destaca un tratamiento localizado de eliminación de malas hierbas con maquinaria real (la flota de RHEA) y en cultivos reales, que se ejecutó con un alto grado de automatización y cuyos resultados fueron validados por expertos satisfactoriamente. De todo esto puede concluirse que el trabajo aporta importantes contribuciones a la Agricultura de Precisión y a la robótica agrícola. Estas

contribuciones han sido publicadas en revistas y presentadas en numerosos congresos científicos internacionales.

Palabras Clave

Planificación de flotas de robots terrestres, supervisión de flotas de robots, gestión/integración de misiones de monitorización/inspección e intervención/tratamiento, optimización combinatoria, meta-heurísticas, operador de vecindad/mutación, Recocido Simulado, Algoritmos Genéticos, NSGA-II, Agricultura de Precisión, automatización de tareas agrícolas, tratamiento localizado de malas hierbas.

Abstract

Safe and effective weed control based on the principles established by Precision Agriculture requires specific and advanced technology at a reasonable cost within profit margins. Using ground and air autonomous mobile platforms, equipped with innovative perception, intelligent decision-making systems and tools that allow precise herbicide application treatments, reduces the cost associated as well as the operation and the potential environmental damage and risk to farmers. Several international research groups are already dedicated to the development of robotic technology to optimize complex operations involved in this type of precision treatments.

The development of autonomous support systems for field and woody crops precise treatments is the main objective of the RHEA project (Robot Fleets for Highly Effective Agriculture and Forestry Management), funded by the 7th Framework Programme EC and addressed to minimize inputs (agrochemicals, fuel, etc.) while ensuring product quality and safety as well as full coverage of the fields regardless of their size. RHEA proposes to use a fleet of both air and terrestrial small/medium robots for inspection/monitoring and subsequent application of pesticides, which has many advantages compared with traditional treatments based on a big machine. For example, a fleet of small/medium robots reduces the impact on soil compaction and interacts more safely with the farm operators, because the detection and interaction can be separated into different detection and fault management systems, as proposed in this thesis.

The research presented in this thesis has been developed within this European project and is related to the design, development and evaluation at the highest level of the RHEA architecture, in other words, to the three fundamental aspects to make the robots of the fleet to execute efficiently the work and without human intervention, i.e. to the planning, the supervision and the complete and integrated management of the treatment task or mission.

Planning is necessary to obtain the paths to be followed by the ground units of the fleet to carry out the task in coordination. In particular, a combinatorial optimization problem is presented in the thesis, whose objective is to determine the routes that optimize the performance of the task in terms of some criteria (cost inputs required time, distance, etc.). The thesis formulates the problem in a very general way to include many real agricultural scenarios. That is, using one or more vehicles simultaneously with both homogeneous and heterogeneous characteristics, with real physical constraints such as turning radius or storage capacity for products (both pesticides and fuel). It also considers the possibility of stopping for refuelling and the variability of the crop, for example, by distribution maps of stands of weeds. In addition, any field shape and even fields composed by several parcels have been considered. Finally, several criteria have been used to determine the best plans for each vehicle of the fleet, either combining them into a single objective or using multi-objective optimization. The formulated optimization problem is solved using and comparing three meta-heuristics: Simulated Annealing, Genetic Algorithms and Non-dominated Sorting Genetic Algorithm II (NSGA-II). In addition, to accelerate the convergence of these methods, a neighbourhood/mutation operator, mix-opt, has been proposed and tested. This new operator builds new solutions based on a weighted combination of well-known operators used for similar problems,

Once the optimum plans for the elements of the fleet have been found, the next step is to run autonomously plans and check that the execution has been carried out as

planned. In autonomous systems it may happen that the behaviour deviates from expected because of its own complexity and, above all, because of the uncertainty of the external factors that affect them, very unpredictable and difficult to control, especially when working outdoors as it is the case addressed in this thesis. Therefore, it is vital to highlight these situations as soon as possible to correct them or even stop them, especially when this involves moving machinery control systems of a certain size which can be potentially dangerous. In this context the thesis proposes a distributed supervision architecture composed of various levels, based on the concepts of supervisor and alarm. These two concepts allow decoupling supervision into independent and easy modules to implement, which can be distributed. Furthermore, since supervisors can be used to monitor other supervisors, they also allow incrementally building a larger overall monitoring system in several layers, first developing low levels dedicated to monitoring the simplest behaviours and then higher levels, dedicated to the control of the most complex behaviours that result from the combination of the lowest levels. Based on this architecture, a multi-level supervising system was designed. The lowest level operates within the ground units and detects their problems individually. A higher level monitors those aspects concerning various elements of the fleet simultaneously (for example, collisions between two or more vehicles). And finally, a third level gathers all the information to report to a human operator and allow him to take control if needed.

Finally, the thesis proposes a management system to achieve a full integration between the monitoring/inspection carried out by the air units of the fleet and the intervention/treatment carried out by the ground units, allowing for a single human operator to define the planning data input, calculate the plans, run them on the units as part of an overall mission and receive an easily understandable monitoring information generated by the supervision. The end result is a system which permits the operator in charge of the fleet to link the different steps to carry out an agricultural treatment task in a totally automated way. Of course, the full automated management of the task/mission contains a system that reports in real time about the status of all the fleet elements involved in the mission.

The three aspects, planning, supervision and management, were evaluated by numerous theoretical and practical tests, including a selective weed treatment with real machinery (RHEA fleet) and real crops, which was executed with a high degree of automation and the results were successfully validated by experts. From all this, it can be concluded that the study provides important contributions to precision agriculture and agricultural robotics. These contributions have been published in international journals and many scientific conferences.

Keywords

Planning for fleets of ground robots, supervising fleet of robots, management/integration of monitoring/inspection and intervention/treatment missions, combinatorial optimization, meta-heuristics, neighbourhood/mutation operator, Simulated Annealing, Genetic Algorithms, NSGA-II, precision agriculture, automation of agricultural tasks, site-specific weed control

Índice general

Resumen.....	XIII
Abstract	XVII
Índice de figuras.....	XXIII
Índice de tablas.....	XXIX
Listado de acrónimos	XXXI
Listado de símbolos	XXXIII
1 INTRODUCCIÓN	1
1.1. Descripción del problema	1
1.2. Proyectos que motivaron la investigación.....	6
1.3. Objetivos	8
1.4. Publicaciones	12
1.4.1. Revistas	12
1.4.2. Capítulos de libros	14
1.4.3. Congresos.....	14
1.5. Estructura de la tesis	20
2 ESTADO DEL ARTE.....	23
2.1. Automatización agrícola	23
2.2. Planificación de la tarea	25
2.3. Supervisión de la ejecución de la tarea	33
2.4. Separación de la tarea agrícola en inspección y tratamiento.....	36
3 PLANIFICACIÓN DE TAREAS AGRÍCOLAS	39
3.1. Formulación: problema de optimización combinatoria	39
3.2. Algoritmos meta-heurísticos	46
3.2.1. Estrategia general de los algoritmos de optimización empleados.....	48
3.2.2. Representación de la solución.....	49
3.2.3. Funciones objetivo	56
3.2.4. Selección de las mejores soluciones	64
3.2.5. Generación de las nuevas soluciones	64
3.2.6. Reemplazando el conjunto de trabajo	72
3.2.7. Recocido Simulado	72
3.2.8. Algoritmos Genéticos	75
3.2.9. Non-Dominated Sorting Genetic Algorithm II	77

3.2.10. Comparativa de los métodos	81
3.3. Obtención de las mejores configuraciones iniciales para los parámetros	81
3.3.1. Configuración del operador mix-opt	85
3.3.2. Configuración de los parámetros del método Recocido Simulado	87
3.3.3. Configuración de los parámetros del Algoritmo Genético.....	88
3.3.4. Configuración de los parámetros del NSGA-II.....	90
3.4. Resultados	93
3.4.1. El operador mix-opt versus los operadores de la literatura.....	93
3.4.2. Estudio comparativo de los métodos de optimización	101
3.4.3. Pruebas ilustrativas	104
3.4.4. Comparación con otros autores	121
4 SUPERVISIÓN DE MISIONES	127
4.1. Arquitectura de supervisión	127
4.1.1. Niveles de supervisión	129
4.1.2. Supervisor de Misión	130
4.1.3. Módulo de Reparación de Fallos.....	132
4.1.4. Gestor de Notificación de Alarmas	132
4.2. Instrumentación.....	133
4.2.1. Percepción: Sensores.....	135
4.2.2. Actuación: Aperos.....	136
4.2.3. Control: Sistema de Control de Unidad	139
4.3. Sistema se supervisión implementado	141
4.3.1. Implementación del Supervisor de Misión.....	141
4.3.2. Implementación del Módulo de Reparación de Fallos.....	146
4.3.3. Implementación del Gestor de Notificación de Alarmas	147
4.4. Resultados	147
4.4.1. Test de misión de pulverización localizada de malas hierbas	149
4.4.2. Test de colisiones	155
5 GESTOR DE MISIONES	161
5.1. Estructura	161
5.1.1. Controladores de misión	164
5.1.2. Despachador.....	170
5.2. Implementación para RHEA.....	171
5.2.1. Interfaz GUI y Comunicaciones entre módulos.....	172
5.3. Resultados	176

6 CONCLUSIONES Y TRABAJO FUTURO	183
6.1. Conclusiones	183
6.1.1. Planificación.....	183
6.1.2. Supervisión.....	185
6.1.3. Gestión	186
6.2. Trabajo futuro	187
6.2.1. Planificación.....	187
6.2.2. Supervisión.....	189
6.2.3. Gestión	190
7 BIBLIOGRAFÍA	191

Índice de figuras

Figura 2.1. Tipos de maniobra para un vehículo de radio r_{min} en un campo regular. (a) Π -giro, (b) Ω -giro y (c) T-giro.....	28
Figura 3.1. Tipos de navegación dentro de cultivo. (a) Vehículo pequeño dentro de cultivo de calle estrecha, (b) vehículo medio dentro de cultivo de calle ancha, (c) vehículo medio con solo ruedas dentro de calles de cultivo de calle ancha y (d) vehículo medio desplazándose en cultivo de calle estrecha en etapa temprana de crecimiento.....	40
Figura 3.2. Campo dividido en 7 pistas de 5 líneas de cultivo cada una.....	41
Figura 3.3. (a) Campos con formas irregulares y con direcciones de cultivo variables, y (b) su representación en pistas.	42
Figura 3.4. Clasificación de los algoritmos meta-heurísticos más habituales. La clasificación es una extensión de la propuesta en (Dréo, 2015) que incluye la categoría multi-objetivo.....	47
Figura 3.5. Procedimiento que resume la parte más externa de funcionamiento de los tres algoritmos meta-heurísticos seleccionados	49
Figura 3.6. Orientaciones de un cultivo	51
Figura 3.7. Las 6 trayectorias posibles según el teorema de Dubins (Dubins, 1957) entre dos puntos con orientación de salida y orientación de entrada fijas.	52
Figura 3.8. Trayectorias asociadas a la permutación $\sigma = (p3, p6, p8, p10, s2, p7, p2, p1, s1, s3, p9, p4, p5)$ y el vector de reabastecimientos $b = (1, 0, 1, 0, 0, 0, 0, 0, 0, 0)$	53
Figura 3.9. Cruce en (a) un punto, (b) dos puntos y (c) uniforme.....	65
Figura 3.10. Cruce PMX. Primero se realiza un cruce en dos puntos y después emparejamientos entre homólogos hasta eliminar las repeticiones.....	66
Figura 3.11. Operadores de mutación típicos en problemas de rutas. (a) <i>Relocate</i> , (b) <i>swap</i> , (c) <i>simple inversion</i> , (d) <i>cut inversion</i> , (f) <i>or-opt</i> y (e) <i>LSM</i>	67
Figura 3.12. Operadores de mutación para el vector de reabastecimientos. (a) Inversión binaria y (b) acción aleatoria.....	68
Figura 3.13. Rango de la función $\exp(-\delta ET)$ cuando δE es positiva (por tanto, cambios a peor) y δET se mueve entre 0 y 10.	73
Figura 3.14. Recocido Simulado con recalentamiento.....	74
Figura 3.15. Algoritmo Genético	77
Figura 3.16. Frentes de Pareto (líneas azul oscuro) para las funciones $f1$ y $f2$ que producen la imagen acotada por una circunferencia, cuando los objetivos son (a) minimizar $f1$ y maximizar $f2$, (b) maximizar $f1$ y maximizar $f2$, (c) minimizar $f1$ y minimizar $f2$, y (d) maximizar $f1$ y minimizar $f2$	78

Figura 3.17. (a) Frentes para un grupo de individuos dispuestos según dos valores objetivos que se deben minimizar, y (b) distancia de crowding del individuo i del primer frente.	79
Figura 3.18. NSGA-II	80
Figura 3.19. El mismo campo cubierto en cuatro casos: con salida y llegada libre por (a) punto de salida y llegada sin fijar y un sólo vehículo, (b) punto de salida y llegada sin fijar y varios vehículos, (c) punto de salida y llegada fijo y un sólo vehículo y (d) punto de salida y llegada fijo y varios vehículos	84
Figura 3.20. Campo utilizado en los procesos de configuración.....	84
Figura 3.21. Valores objetivo en media (10 ejecuciones por configuración) para las 9.720 configuraciones consideradas del operador mix-opt.....	86
Figura 3.22. Valores objetivo medios, obtenidos en el proceso de configuración del Recocido Simulado	88
Figura 3.23. Resultados medios obtenidos en el proceso de configuración del Algoritmo Genético considerado.....	89
Figura 3.24. Valores objetivo medios normalizados, obtenidos en el proceso de configuración del NSGA-II para los criterios (a) coste por insumos y (b) tiempo de tarea.....	91
Figura 3.25. Población de soluciones optimizadas en coste por insumos y tiempo simultáneamente.....	92
Figura 3.26. Clasificación de los operadores para cada problema CVRP según sus resultados medios.....	95
Figura 3.27. <i>Gaps</i> para cada operador en cada problema CVRP.....	97
Figura 3.28. <i>Gaps</i> medios de los operadores para 1, 7,5, 15, 30 y 60 minutos.....	97
Figura 3.29. Convergencia de los operadores de la literatura, el operador mix-opt y sus variantes en un problema de planificación agrícola. Ambos ejes están en escala logarítmica.	100
Figura 3.30. Campo de 54 ha con infestación de un 30 %. Mapa de distribución de mala hierba generado aleatoriamente.....	102
Figura 3.31. Convergencia media de cada algoritmo para cada criterio de optimización.....	102
Figura 3.32. Rutas obtenidas en el test 1 usando un único vehículo y considerando distintos radios de giro: (a) 1 m, (b) 3 m y (c) 4 m.	105
Figura 3.33. Rutas del test 2 usando varios vehículos y varios radios de giro: (a) 1 m, (b) 3 m y (c) 4 m.	107
Figura 3.34. Rutas del test 3 usando una flota de 3 vehículos con tanques de 10, 110 y 1110 dosis y una distribución concreta de mala hierba.....	109
Figura 3.35. Rutas obtenidas en el test 4 para una flota de robot con iguales características y optimizando distintos criterios: (a) coste por insumos, (b) tiempo de tarea y (c) ambos.....	110

Figura 3.36. Rutas obtenidas en el test 5 para una flota de robots con iguales características y optimizando distintos criterios: (a) coste por insumos, (b) tiempo requerido para la realización de la tarea y (c) los criterios anteriores simultáneamente.....	114
Figura 3.37. Rutas obtenidas en las condiciones del test 5 usando un coste por reabastecimiento igual al coste de arranque de un vehículo, es decir, en ambos casos igual a 0,2 €.....	115
Figura 3.38. Rutas obtenidas en el test 6 para una flota de robots con distintas velocidades y optimizando distintos criterios: (a) coste por insumos, (b) tiempo requerido para la realización de la tarea y (c) los criterios anteriores simultáneamente.....	117
Figura 3.39. Ruta obtenida que minimiza el coste por insumos cuando los costes por desplazamiento son los que se muestran en la Tabla 3.26.....	118
Figura 3.40. Rutas obtenidas para el test 7 usando costes contrapuestos para el coste de desplazarse y el coste de pulverizar	120
Figura 3.41. Rutas obtenidas para el test 8 usando una flota de 3 vehículos homogéneos y minimizando el tiempo de tarea.	121
Figura 3.42. Rutas obtenidas para el primer campo, según la solución de (a) Bochtis y (b) el Planificador propuesto.....	122
Figura 3.43. Soluciones obtenidas por (Bochtis & Vougioukas, 2008) para (a) el campo2 y (c) campo3. Rutas obtenidas por el Planificador desarrollado para (b) el campo2 y (d) campo3.....	123
Figura 3.44. Soluciones obtenidas para el campo4 por (a) (Bochtis & Vougioukas, 2008) y (b) el Planificador desarrollado.....	124
Figura 3.45. Comparación de las rutas obtenidas para el campo extraído de (Hameed, Bochtis, & Sorensen, 2011). (a) Ruta obtenida en el artículo de referencia y (b) ruta generada por el Planificador propuesto y desarrollado en este trabajo de investigación.	126
Figura 4.1. Supervisión distribuida y multi-nivel	129
Figura 4.2. Arquitectura de supervisión donde se muestra el Supervisor de Misión (nivel 2) en detalle. Los rectángulos representan los supervisores del nivel 2 organizados jerárquicamente y las flechas las entradas y las salidas, donde los mensajes pueden ser tanto alarmas como información de monitorización.	131
Figura 4.3. Arquitectura del proyecto RHEA	133
Figura 4.4. Cabina del tractor antes (a) y después de la adaptación (b).....	134
Figura 4.5. Flota de RHEA: (1) Sistemas de percepción, (2) barra de pulverización, (3) herramienta mecánico-térmica, (4) barra de pulverización de inyección directa y (5) sistemas de control.	134
Figura 4.6. Caja con cámara, IMU, termostato y ventilador.....	135
Figura 4.7. Dispositivos instalados en los tractores	136
Figura 4.8. Equipo de pulverización de herbicida desarrollado por Agrosap para RHEA.	137
Figura 4.9. Herramienta mecánico-térmica para la eliminación precisa de malas hierbas.	138

Figura 4.10. Pulverizador de fitosanitario por aire comprimido.....	138
Figura 4.11. Avance incremental de las posiciones del <i>supervisor de colisiones</i>	144
Figura 4.12. Superficie que estima el <i>supervisor de colisiones</i> que ocupa una unidad	145
Figura 4.13. (a) Campo con los rodales dibujados con líneas blancas. (b) Representación matricial del campo; mapa de distribución de malas hierbas.....	148
Figura 4.14. Trayectoria y estado de las boquillas durante la misión.	149
Figura 4.15. Alarmas generadas por el <i>supervisor de la velocidad</i> en función de la velocidad real, la esperada y el margen (umbral) de error utilizado. (a) Alarmas generadas a lo largo del tiempo para umbrales de 0,5 km/h, 1 km/h y 1,5 km/h. (b) Alarmas generadas a lo largo de la trayectoria para un umbral de error de 0,5 km/h	151
Figura 4.16. Alarmas generadas por el <i>supervisor de estado del apero</i> en función del estado real y el esperado durante (a) la misión completa o recorrido completo del campo y (b) en el intervalo que transcurre entre los segundos 285 y 321.....	152
Figura 4.17. Alarmas generadas por el <i>supervisor de trayectoria</i> en función de la trayectoria real y la esperada.....	153
Figura 4.18. Información generada por el <i>supervisor de misión restante</i>	153
Figura 4.19. Alarmas generadas por el <i>supervisor de disrupción de servicios</i>	154
Figura 4.20. Alarmas generadas por el <i>supervisor de unidad</i> a lo largo del tiempo	154
Figura 4.21. Alarmas generadas por el <i>supervisor de unidad</i> a lo largo de la trayectoria.	155
Figura 4.22. Trayectorias de los dos vehículos en la segunda misión.....	156
Figura 4.23. Cronología de la misión: colisiones resueltas mediante pausas (a, c, e, g, i, k), reanudaciones de la marcha (b, d, f, h, j, l) y puntos finales (m, n).	157
Figura 4.24. Alarmas generadas y evolución de los estados internos que el <i>supervisor de colisiones</i> usa durante la misión.....	159
Figura 4.25. Salidas y entradas del <i>supervisor de misión completada</i>	159
Figura 5.1. Estructura interna del Gestor de Misiones planteado y sistemas externos con los que puede interactuar.....	162
Figura 5.2. Diagrama de estados necesario para poder activar el apero en una unidad desde cualquiera de los estados principales (a) con una representación tradicional y (b) usando la representación <i>statechart</i>	165
Figura 5.3. Diagrama de estados de un controlador de misiones con flotas de vehículos	166
Figura 5.4. Diagrama de estados del controlador de una unidad en misión.....	168
Figura 5.5. Diagrama de estados del controlador básico de unidad.....	169
Figura 5.6. Diagrama de estados del despachador que incluye todas las peticiones que se han considerado.....	170
Figura 5.7. (a) Drone AR200 en vuelo y (b) cámaras con las que estaban equipadas los drones.....	171

Figura 5.8. Interfaz GUI. (a) Panel para realizar peticiones (planificaciones, mosaicados, lanzar misiones, etc.), visualizar la información de monitorización de cada unidad (posición, velocidad, estado del apero) y tele-operar una unidad durante una misión. (b) Panel para visualizar las imágenes proporcionadas por la unidad durante la misión.....	174
Figura 5.9. Interfaz del Planificador: (a) Panel con acceso a todos los parámetros a configurar y (b) panel de evolución de la convergencia	175
Figura 5.10. Interfaz del Planificador. Panel que muestra el resultado final de la planificación.....	176
Figura 5.11. Cultivo de trigo de invierno donde se han protegido con unas lonas de plástico 9 rodales de malas hierbas que se habían sembrado para realizar las pruebas.	177
Figura 5.12. Misión aérea de inspección: (a) planificación y (b) ejecución con rutas reales....	178
Figura 5.13. Generación del mapa de malas hierbas: (a) imagen mosaicada y (b) malas hierbas mapeadas sobre el campo	178
Figura 5.14. (a) Tiras de papel dispuestas a lo largo del campo para verificar la pulverización y (b) rodal real versus rodal sembrado.	179
Figura 5.15. (a) Mapa de malas hierbas empleado para la misión de tratamiento realizada por las unidades terrestre y (b) trayectoria de la unidad con la superficie que pulverizó	180
Figura 5.16. (a) Precisión en la apertura y cierre de las boquillas de pulverización de herbicida respecto al área objetivo. (b) Diferencias en las dos pruebas realizadas entre trayectoria real seguida por el vehículo y la trayectoria establecida.....	181
Figura 6.1. Transiciones válidas en (a) un campo irregular y en (b) un campo formado por varias parcelas conectadas por caminos.....	188

Índice de tablas

Tabla 1.1. Ventajas de una solución basada en una flota de vehículos frente a los sistemas tradicionales con un único gran vehículo.....	4
Tabla 1.2. Superficies y producción de varios tipos de cultivo en la Unión Europea de los 28 en 2014 según la Comisión Europea para el Desarrollo de la Agricultura (Agriculture in the European Union - Markets Statistical Information 2014, 2014).....	11
Tabla 3.1. Plan del vehículo 1 codificado en la solución de la Figura 3.8.....	54
Tabla 3.2. Recursos utilizados por cada operación.....	59
Tabla 3.3. Costes estimados por operación y por criterio de optimización para tractores de 50 CV.....	61
Tabla 3.4. Número mínimo de operaciones que se ejecutan por tarea.....	62
Tabla 3.5. Número máximo de operaciones que se ejecutan por tarea.....	63
Tabla 3.6. Variables de configuración del nuevo operador.....	71
Tabla 3.7. Comparativa Recocido Simulado, Algoritmo Genético y NSGA-II.....	81
Tabla 3.8. Valores de configuración examinados para las variables del operador mix-opt.....	85
Tabla 3.9. Valores de configuración probados para el Recocido Simulado.....	88
Tabla 3.10. Valores de configuración examinados para el Algoritmo Genético.....	89
Tabla 3.11. Costes para la flota en el proceso de configuración del NSGA-II.....	90
Tabla 3.12. Valores analizados para cada variable del método NSGA-II.....	90
Tabla 3.13. Resultados en media obtenidos por cada operador para cada uno de los 10 problemas CVRP analizados. Las filas en gris resaltan los resultados del operador mix-opt y sus variantes.....	95
Tabla 3.14. Resultados en media ordenados y mejoras respecto al operador de la literatura que obtuvo mejores resultados (<i>simple inversion</i>). En gris el mix-opt y sus variantes.....	96
Tabla 3.15. <i>Gaps</i> medios ordenados y mejoras para cada operador respecto al mejor operador de la literatura para 60, 30, 15, 7,5 y 1 minutos. En gris se resaltan los resultados del operador mix-opt y sus variantes.....	98
Tabla 3.16. Resultados medios obtenidos para las 50 ejecuciones de 24 horas del Recocido Simulado realizadas para cada operador. En gris se han resaltado los resultados del operador mix-opt y sus variantes.....	101
Tabla 3.17. Mejores resultados por método y criterio a lo largo del tiempo.....	103
Tabla 3.18. Costes usados en el test 1, ajustados para calcular solo la distancia en cabecera.....	106
Tabla 3.19. Distancias y soluciones del test 1 según el radio de giro considerado.....	106

Tabla 3.20. Distancias y soluciones del test 2 según el radio de giro, y comparadas con las distancias obtenidas en el test 1.	108
Tabla 3.21. Características de los vehículos usados en el test 4	110
Tabla 3.22. Coste de cada solución obtenida para cada criterio de optimización en el test 4. Solo se muestra el vector de pistas de las soluciones porque no fue necesario ningún reabastecimiento para ninguno de los tres casos.....	112
Tabla 3.23. Coste de cada solución obtenida para cada criterio de optimización en el test 5...	115
Tabla 3.24. Costes en tiempo de las operaciones de desplazamiento usadas en el test 6	116
Tabla 3.25. Costes de cada solución obtenida para cada criterio de optimización en el test 6. Solo se muestra el vector de pistas de las soluciones porque no fue necesario ningún reabastecimiento para ninguno de los tres casos.....	116
Tabla 3.26. Costes por desplazamiento según vehículo.....	118
Tabla 3.27. Costes considerados por insumos para los vehículos de la flota en el test 7.....	119
Tabla 3.28. Coste por insumos de tratar una pista según el número de malas hierbas que contenga y el vehículo que se emplee.....	119
Tabla 3.29. Soluciones (vectores permutación) obtenidas por ambos métodos para cada campo.	124
Tabla 3.30. Distancias obtenidas por las soluciones obtenidas por ambos métodos (véase Tabla 3.29) para cada campo.	125
Tabla 4.1. Información de monitorización que proporciona periódicamente el SCU.....	139
Tabla 4.2. Alarmas que puede generar el SCU para el nivel 2 (Supervisor de Misiones).	140
Tabla 4.3. Ordenes más importantes que genera el SCU para los dispositivos de a bordo, incluido el vehículo.....	141
Tabla 4.4. Alarmas generadas por los supervisores de nivel básico.	143
Tabla 4.5. Alarmas del <i>supervisor de flota</i>	143
Tabla 4.6. Relaciones alarma-actuación del Módulo de Reparación de Fallos.....	146
Tabla 4.7. Plan de actuación del vehículo hasta la finalización del primer rodal (véase trayectoria de la Figura 4.14).	150
Tabla 5.1. Operaciones básicas proporcionadas por las unidades y que se han considerado para diseñar los controladores de misión.	166
Tabla 5.2. Formato de mensaje	172
Tabla 5.3. Ejemplo de mensaje con carácter de escape sustituido.....	173

Listado de acrónimos

ACK	Acknowledgement - Mensaje de confirmación de recepción
BD	Base de Datos
CPP	Chinese Postman Problem
CVRP	Capacited Vehicle Routing Problem
GNSS	Global Navigation Satellite System
GUI	Graphical User Interface
GPS	Global Positioning System
LiDAR	Light Detection And Ranging
LSM	Local Search Mutation
MFVRP	Mixed Fleet Vehicle Routing Problem
mTSP	Multi Traveling Salesman Problem
NP (Problema)	Nondeterministic Polynomial time problem
NSGA-II	Non-dominated Sorting Genetic Algorithm II
PMX	Partially Mapped Crossover
Qt	Biblioteca multiplataforma para desarrollar interfaces gráficas
RHEA (Proyecto)	Robot Fleets for Highly Effective Agriculture and forestry management
RTK	Real Time Kinematic
SCU	Sistema de Control de Unidad
TSP	Traveling Salesman Problem
VRP	Vehicle Routing Problem

Listado de símbolos

α	Variable de configuración del Recocido Simulado. Factor de enfriamiento
a	Variable de configuración del mix-opt. Número de intentos de la búsqueda local
b	Vector binario de reabastecimientos
c_i^k	Coste de tratar la fila i con el vehículo k
c_{ij}^k	Coste de transitar la fila i a la j con el vehículo k
d_i	Demanda de la pista i
D_k	Demanda que puede satisfacer el vehículo k con el tanque lleno
e	Variable de configuración del Algoritmo Genético. Tipo de elitismo
l	Variable de configuración del mix-opt. Longitud máxima de los segmentos utilizados en las operaciones elementales
m	Número de vehículos de la flota ($ V $)
m_i	Número de operaciones de tipo i llevadas a cabo
n	Número de pistas en el campo ($ P $)
n_h	Variable de configuración del NSGA-II. Tamaño de la población hija
n_p	Variable de configuración del Algoritmo Genético y el NSGA-II. Tamaño de la población de trabajo
Ω -giro	Maniobra de giro de tipo Ω
Π -giro	Maniobra de giro de tipo Π
p_c	Variable de configuración del Algoritmo Genético y el NSGA-II. Probabilidad de cruce
p_m	Variable de configuración del Algoritmo Genético y el NSGA-II. Probabilidad de mutación
P	Conjunto de los índices de las pistas del campo T
$P(\delta E, T)$	Probabilidad de aceptar un cambio δE cuando la temperatura es T
r	Variable de configuración del mix-opt. Opción que indica si se prueban también los segmentos invertidos en las operaciones elementales
r_{min}	Mínimo radio de giro de un vehículo
σ	Parte de la solución x que contiene una permutación de las pistas del campo
σ_k	Pistas de la permutación σ asociadas al vehículo k
$\sigma(i)$	Pista en la posición i en la permutación σ
$\sigma(i:j)$	Sub-permutación formada por las pistas de σ entre las posiciones i y j
s_i	Separador que marca donde acaban las pista del vehículo i en σ y donde empiezan las del vehículo $i + 1$
S	Conjunto de los índices de los separadores en σ
t_e	Variable de configuración del Recocido Simulado. Tiempo de estancamiento antes de recalentar
T	Temperatura máxima en el Recocido Simulado
T -giro	Maniobra de giro de tipo T
V	Conjunto de los índices de los vehículos de la flota
w_i	Variable de configuración del mix-opt. Peso de la operación elemental i
x	Solución genérica
x^*	Solución óptima
θ_c	Orientación de un cultivo en el sentido cabecera inferior – cabecera superior

1 INTRODUCCIÓN

1.1. Descripción del problema

La gestión agrícola ha mejorado mucho en las últimas décadas gracias a las nuevas tecnologías. El uso de sensores, actuadores, sistemas de posicionamiento global (GNSS), sistemas de información geográfica (GIS) o imágenes de alta resolución, entre otras muchas técnicas, ha permitido automatizar en mayor o menor medida las labores agrícolas y aumentar notablemente la productividad y la calidad de las mismas.

Como resultado, el concepto de Agricultura de Precisión se ha consolidado y se ha convertido en una metodología cada vez más habitual e indispensable para la viabilidad de muchas prácticas agrícolas. Este término, acuñado en los últimos años, se puede definir como la aplicación de las tecnologías y principios disponibles para manejar la variabilidad espacial y temporal relacionada con cualquiera de los aspectos de la producción agrícola con el objetivo de mejorar el rendimiento de la cosecha y la calidad medioambiental (Pierce & Nowak, 1999). En otras palabras, utilizar tanto la tecnología como el conocimiento disponibles para extraer el mayor rendimiento posible en los cultivos con el mínimo impacto medioambiental, gestionando de forma individualizada las distintas áreas de un mismo cultivo en lugar de todo el cultivo por igual como una unidad homogénea.

Las plagas (patógenos de plantas, malas hierbas, invertebrados, etc.) reducen significativamente la producción agrícola. Se estima que son responsables de una reducción de aproximadamente el 40 % (Oerke, Dehne, Schönbeck, & Weber, 1994) en la producción potencial mundial mediante enfermedades (patógenos), compitiendo con las plantas de cultivo por el agua, los nutrientes y la luz (malas hierbas), utilizando el propio cultivo como alimento (invertebrados, conejos, etc.). Por esta razón, entre los muchos nichos o escenarios a explotar en la Agricultura de Precisión, uno de los más prometedores es la aplicación eficiente y precisa de agroquímicos (fitosanitarios y fertilizantes). Solo en Europa se gastaron alrededor de 360.000 toneladas de fitosanitarios para acabar con las plagas en el año 2013 (EUROSTAT, 2013). Se calcula que por cada euro que se invierte en fitosanitarios, se ganan cuatro por el cultivo que deja de perderse (Pimentel, y otros, 1992), y como las ventas de fitosanitarios en Europa ascienden a cerca de 7.700 millones de euros (ECPA, 2012), las ganancias que generan se estiman en más de 30.000 millones cada año solo en ese continente. No obstante, estas estimaciones no tienen en cuenta el gran impacto negativo que estos químicos producen en el

medioambiente, impacto que no es en absoluto despreciable. Por ejemplo, se calcula que los fitosanitarios suponen unos 8.000 millones de dólares anuales en daños medioambientales solo en EEUU (Pimentel, y otros, 1992). Esto se debe principalmente al uso altamente ineficiente que se hace de ellos, ya que solo una cantidad muy pequeña suele alcanzar el objetivo al que están dirigidos. Por ejemplo, en (Miller G. T., 2004) se estima que tan solo el 2 % de los insecticidas por pulverización y el 5 % de los herbicidas alcanzan la especie objetivo, quedándose los porcentajes restantes en el medio en forma de contaminación. Es por esto que la Agricultura de Precisión tiene en este caso un escenario idóneo para reducir costes y contaminación enormemente, a través de la aplicación localizada del agroquímico o incluso utilizando como alternativa sistemas físicos de control de malas hierbas menos contaminantes pero no siempre aplicables a cualquier cultivo. Por ejemplo, utilizando quemadores para acabar con la mala hierba en las líneas de cultivo de maíz a partir de pequeños golpes de calor (Raffaelli, Martelloni, Frascioni, Fontanelli, & Peruzzi, 2013), estrategia que solo es válida en cultivos capaces de resistir el calor como el maíz, ajos, zanahorias, etc.

En definitiva, un ejemplo claro de ineficiencia en el empleo de agroquímicos es el uso que se hace de los herbicidas. Los herbicidas representan prácticamente el 36 % del fitosanitario usado en Europa, unas 128.000 toneladas (EUROSTAT, 2013). Como sucede con el resto de fitosanitarios, la práctica habitual consiste en aplicarlos en todo el campo, sin discriminar entre zonas infestadas y no infestadas, lo que produce un gasto innecesario y aumenta los efectos no deseados en el agua y el ecosistema circundante. Tanto es así que muchos estudios europeos han estimado que en cultivos extensivos o arables sería posible reducir en torno a un 70 % la cantidad de herbicida utilizado. Por ejemplo, (Timmermann, Gerhards, & Kühbauch, 2003) logran reducir la cantidad de herbicida en un 54 % en algunos cultivos alemanes mientras que (Ruiz, Escribano, & Fernández-Quintanilla, 2006) alcanzan una reducción de 71 % en campos de cebada en España.

Aunque a priori pueda parecer que este tipo de tratamiento selectivo no será rentable por el alto precio de la maquinaria necesaria (vehículos adaptados, sensores, aperos más sofisticados, etc.), en EEUU se ha demostrado experimentalmente que sus beneficios netos compensan los costes asociados (Rider, Vogel, Dille, Dhuyvetter, & Kastens, 2006).

Todas estas cifras y estudios son una buena muestra de los beneficios asociados a hacer una gestión del cultivo siguiendo los preceptos de la Agricultura de Precisión. Los beneficios se pueden resumir en tres puntos:

1. Incremento del margen de ganancia económica, ya sea aumentando el rendimiento de la cosecha o reduciendo los insumos, fundamentalmente fitosanitarios y fertilizantes.
2. Reducción de la contaminación medioambiental (suelo y aguas subterráneas) debida a los agroquímicos que no alcanzan la diana.
3. Proceso de mayor calidad por dirigirse con precisión a objetivos concretos teniendo en cuenta las características propias de cada campo y tarea.

Aunque los beneficios de una gestión agrícola de precisión son claros, a tenor de los múltiples estudios realizados, el desarrollo tecnológico que requiere este tipo de agricultura para ser implantada con garantías en la práctica diaria todavía tiene mucho recorrido; de hecho, no existen soluciones comerciales que permitan la integración

completa de las prácticas de precisión en una explotación agrícola, aunque sí hay muchas soluciones disponibles que permiten mejorar la precisión a la hora de realizar tareas o aspectos muy concretos, por ejemplo, los sistemas de navegación (John Deere Guidance Systems, 2015) o los aperos que ajustan su altura y su inclinación automáticamente (Pulverizador remolcado serie M700/M700i John Deere) dependiendo del cultivo.

Entre las tecnologías imprescindibles a incluir se encuentra la navegación autónoma, absolutamente indispensable para que las plataformas puedan mover los sensores y actuadores por todo el cultivo. En este sentido existen ya algunos prototipos comerciales, por ejemplo, el tractor Spirit (Autonomous Tractor Corporation, 2015), el sistema de guiado GuideConnect (Fendt GuideConnect, 2011), los sistemas de autonavegación de CIH (Sistemas agrícolas avanzados CIH, 2015) y John Deere (John Deere Guidance Systems, 2015), y los pequeños robots agrícolas de Robotnik (Robotnik, 2015). Sin embargo, estas propuestas en algunos casos sólo resuelven el problema parcialmente al ser los vehículos semiautónomos, navegando de modo autónomo en el cultivo y realizando los giros en cabecera manualmente, o no pasan de ser prototipos capaces de realizar en el campo tareas muy sencillas que requieren poca automatización, centrando su cometido en la navegación autónoma. Igualmente, en el terreno académico existen numerosos trabajos al respecto, pero también están pensados para tareas que no requieren actuación tales como inspecciones autónomas del cultivo (Ruckelshausen, y otros, 2009) (Weiss & Biber, 2011).

Llevar a cabo autónomamente tareas más complejas es mucho más difícil por el mero hecho de que existen multitud de operaciones diferentes (arar, pulverizar, cosechar, sembrar...) todas con sus propios requisitos, hasta el punto de que pueden aparecer implementaciones muy ineficientes si se aborda el problema de una forma muy general, por ejemplo, construyendo una gran plataforma capaz de integrar todas las herramientas para dar respuesta a todas las tareas posibles. De ahí que la posibilidad de reconfigurar el equipamiento de a bordo de las plataformas surja como una propuesta atractiva. En definitiva, sería interesante que las plataformas fueran genéricas, es decir, que pudieran equiparse con unas herramientas u otras y se reconfigurasen automáticamente. De este modo se dispondría de plataformas más versátiles, válidas para más tareas y tipos de cultivo.

Para llevar a cabo esta propuesta es conveniente analizar la relación que existe entre las características de las plataformas móviles y las características de los cultivos que deben tratar. Por un lado es deseable que las plataformas sean lo suficientemente grandes para realizar la tarea en un tiempo razonable y, por otro, lo más pequeñas posible para que sean más livianas y con ello disminuya la compactación que producen en el suelo; un efecto devastador para el cultivo porque elimina el aire y el agua necesarios para el crecimiento de las plantas (Hamza & Anderson, 2005). De todo esto surge la propuesta de emplear flotas de vehículos robotizados de tamaño medio/pequeño capaces de cooperar y de complementarse para ejecutar colaborativamente la tarea agrícola. Las ventajas de esta solución sobre la utilización de un único gran vehículo equipado con muchos actuadores y herramientas diferentes son muchas y se resumen en la Tabla 1.1.

Las plataformas o robots móviles no tienen por qué limitarse a unidades de tierra con sistemas de percepción y actuación a bordo. También es interesante considerar vehículos aéreos tipo drone que, aunque en su mayoría no tienen capacidad para transportar actuadores, sí pueden equiparse con sensores como cámaras (visible, infrarroja, multispectral, etc.) y mejorar la tarea de inspección al poder realizarla desde una perspectiva cenital, con mayor cobertura, o lo que es lo mismo en un tiempo menor, y

sin dañar el cultivo, lo que explica el gran auge que ha sufrido esta tecnología en los últimos años en general y en particular en agricultura (Doering, 2014). Una forma de integrar drones en tareas agrícolas consiste en combinar tareas de inspección mediante unidades aéreas baratas y ligeras, con tareas de tratamiento a cargo de unidades de tierra, más potentes y con capacidad para transportar las herramientas de actuación necesarias. Por ejemplo, usando los drones para realizar un mapa de la distribución de los rodales de malas hierbas en el cultivo (Torres-Sánchez, López-Granados, De Castro, & Peña-Barragán, 2013) y suministrar esta información a los tractores para que traten los rodales de forma localizada. A pesar de los claros beneficios de esta combinación, hasta la fecha ningún proyecto hasta el proyecto europeo RHEA se ha centrado en implementar todo el proceso (inspección-mapa-tratamiento) de forma totalmente automatizada.

Tabla 1.1. Ventajas de una solución basada en una flota de vehículos frente a los sistemas tradicionales con un único gran vehículo

	Un gran tractor/plataforma	Flota de robots pequeños/medios
Productividad	Estándar	La misma o mejor que la estándar porque depende del número de vehículos en la flota
Seguridad en modo de trabajo autónomo	Problema grave de seguridad para los operarios en caso de fallo	Por su tamaño, los robots pequeños/medios interactúan de una forma más segura con los operarios
Tolerancia a fallos de la tarea	Un fallo para la ejecución de la tarea hasta que se repara la máquina	Si falla un vehículo, es posible re-planificar para que el resto de la flota realice la tarea que deja pendiente
Adaptación a una tarea nueva concreta	Dependiendo del tipo de tarea y cultivo no siempre es fácil adaptar un gran vehículo.	Más fácil adaptar/integrar un nuevo sensor/actuador en un robot pequeño/medio que en una gran máquina. Además, si es necesario, existe la posibilidad de adaptar solo algunos vehículos de la flota
Impacto en el campo	Alto nivel de compactación del terreno	Menor compactación debido fundamentalmente al menor peso de los vehículos
Personal	Un operario por vehículo	Un operario para toda la flota

Por todo lo expuesto, es obligado explorar la conveniencia de una alternativa al modelo tradicional agrícola que utilice flotas de robots pequeños, seguros, reconfigurables, heterogéneos y complementarios. Además es conveniente empezar automatizando las tareas de inspección y tratamiento de malas hierbas y gestión de fitosanitarios, ya que, por el volumen y la magnitud de las cifras mostradas, parece el escenario más propicio para alcanzar soluciones rentables.

Para poner a punto una flota de vehículos autónomos que realicen eficazmente tareas agrícolas, hay que superar varios retos a alto nivel, siendo los principales la planificación de las trayectorias a seguir por los robots y la supervisión de las misiones para verificar que la tarea se ejecuta según lo previsto o planificado. Asimismo, también es importante el tratamiento de los datos adquiridos o generados durante la ejecución de la misión y su gestión para poder reutilizarlos en nuevas misiones y así llevar a cabo tareas más complejas compuestas por varios pasos o fases.

Dejando de lado los robots aéreos, que no son el objeto de esta tesis, y centrándonos en las habituales plataformas agrícolas de trabajo (tractores, cosechadoras, etc.), si planificar las trayectorias de trabajo para un único vehículo agrícola tradicional no autónomo es ya una cuestión esencial porque afecta directamente a aspectos tan importantes como el combustible consumido y la compactación del suelo, planificar para una flota de vehículos autónomos resulta aún más importante si se repara en que hay que considerar el potencial riesgo de colisión. Si además los vehículos no son homogéneos, se hace todavía más difícil la planificación con una distribución adecuada de tareas entre vehículos, ya que los planes óptimos pueden variar radicalmente dependiendo de las características particulares de cada uno de ellos. Por ejemplo, factores como la velocidad de trabajo, la capacidad de carga o el radio de giro pueden influir en la selección de las zonas más adecuadas para destinar cada vehículo. Además, otros elementos como la forma del campo, las propias características de la tarea (cobertura parcial o total) o el criterio de minimización (tiempo, combustible, distancia) también pueden influir en los planes.

Por otro lado, una vez encontrados los planes óptimos que incluyen las rutas óptimas, es necesario supervisar que la misión se ejecuta según lo planificado para garantizar un tratamiento de calidad. Al trabajar la flota en exteriores con condiciones muy difíciles de controlar, tales como cambios en el terreno, inclemencias meteorológicas, la aparición repentina de personas o animales en la escena, etc., no se puede garantizar con total seguridad que las misiones se ejecuten exactamente según lo previsto, lo que no solo supone un problema de rendimiento sino también de seguridad. Por ello, es necesario supervisar estrechamente la actividad de todos los elementos de la flota durante la ejecución de una misión, para poder detectar lo más rápidamente posible los problemas que puedan surgir, y, si es posible, subsanarlos cuanto antes.

Por último, para alcanzar la plena automatización, es necesario algún tipo de sistema global que permita al operario definir como entrada los datos del campo y la tarea que quiere realizar, y que a partir de ahí dirija el flujo de trabajo. En otras palabras, un sistema Gestor que permita enlazar los diferentes pasos para llevar a cabo una tarea agrícola de forma totalmente automatizada y que, por supuesto, informe al operario a cargo de la flota de todo lo que va sucediendo durante la misión. Este operario podrá, asimismo, tomar el control de los elementos de la flota siempre que aparezca un problema que el sistema de supervisión no sepa resolver.

Éstas son las tres cuestiones que se tratan a fondo en esta tesis: planificación, supervisión y gestión de los pasos necesarios para completar tareas agrícolas mediante flotas de robots de forma totalmente autónoma. A lo largo del documento se irán tratando estos temas aportando estrategias generales que en muchos casos se centrarán en el tratamiento localizado de malas hierbas con herbicidas por su vital importancia.

1.2. Proyectos que motivaron la investigación

La investigación asociada a esta tesis tuvo su origen y marco de trabajo en los proyectos científicos de ámbito europeo y nacional que se citan a continuación:

- *Robot Fleets for Highly Effective Agriculture and Forestry Management (RHEA)*, 2010-2014). Proyecto europeo (NMP-CP-IP 245986-2) dentro del VII Programa Marco de Investigación y Desarrollo tecnológico (FP7) desarrollado durante los años 2010-2014 por el consorcio formado por el Centro de Automática y Robótica del CSIC, el Instituto de Ciencias Agrarias del CSIC, la Universidad Politécnica de Madrid, la Universidad Complutense, Case New Holland, AirRobot, Cyberbotics, Bluebotics, GogVis, Agrosap, Forschungszentrum Telekommunikation Wien, el Instituto de Agricultura Sostenible del CSIC, Irstea, la Universidad de Pisa, la Universidad de Florencia, CM Srl y Tropical. La finalidad del proyecto era el diseño, desarrollo y prueba de una nueva generación de sistemas automáticos y robóticos para la eliminación precisa y eficaz de malas hierbas y plagas en entornos agrícolas y forestales mediante medios químicos y físicos (mecánicos y térmicos). Los escenarios posibles que contemplaba eran muy variados por abarcar los principales cultivos europeos ya fueran de calle estrecha (cereales de invierno como el trigo y la cebada), de calle ancha (tomate, maíz, fresa, girasol y algodón) o leñosos perennes (almendros, nogales, olivos, álamos, etc.). Entre los objetivos de RHEA se encontraba la disminución del uso de agroquímicos en agricultura en un 75 %, lo que conduciría a mejorar la calidad de los cultivos, la salud y la seguridad de los seres humanos, y a la reducción de los costos de producción a partir de una gestión sostenible. RHEA proponía el desarrollo de una flota de pequeños robots, terrestres y aéreos, equipados con sensores avanzados y aperos mejorados y más precisos) así como el desarrollo de algoritmos sofisticados de toma de decisiones.
- *Sistema de inspección terrestre en vehículos autónomos y su aplicación efectiva a la detección de malas hierbas y su control localizado (GroW)*. Proyecto del Plan Nacional de I+D+i (AGL2011-30442-C02-02) coordinado durante los años 2011-2014 por el Centro de Automática y Robótica (CSIC-UPM) y en el que participó la Universidad Complutense de Madrid. El objetivo principal de este proyecto era el desarrollo de técnicas eficientes y efectivas de inspección y discriminación desde tierra de malas hierbas. Entre otras cosas se abordaba la automatización de la recogida de información en campo integrando equipos de inspección en plataformas móviles. Se analizaron distintas estrategias de inspección como la integración de los sensores (fundamentalmente cámaras) en tractores convencionales de forma que se pudiesen tomar imágenes a la vez que se realizaban otras labores en el campo o la inspección totalmente autónoma en la que se dispone de un robot móvil con una dotación sensorial adecuada, expresamente diseñado para la tarea de inspección.
- *Percepción visual, toma de decisión y actuación en sistemas artificiales complejos: Control selectivo de malas hierbas en cultivos agrícolas (PERICIA)*. Proyecto del Plan Nacional de I+d+i (AGL2008-04670-C03-02) coordinado durante los años 2009-2011 por el Centro de Automática y Robótica del CSIC y en el que participó la Universidad Complutense de Madrid. La finalidad principal de este proyecto era el análisis y desarrollo de nuevos procedimientos para detectar las malas hierbas

presentes en cultivos de maíz y controlarlas de forma localizada con un uso mínimo de herbicidas. En este proyecto, un sistema autónomo de aplicación localizada de herbicidas se concebía como un sistema artificial complejo formado por tres subsistemas: a) un subsistema de percepción o detección e identificación de estructuras naturales, como son las líneas de cultivo y las malas hierbas; b) un subsistema de toma de decisión o elaboración de un plan de actuación, la cantidad de herbicida a aplicar, considerando lo percibido, la información adicional disponible (tipo de cultivo y mala hierba, experiencia sobre el campo, etc.) y los objetivos establecidos; y c) un subsistema de actuación o puesta en marcha del plan de tratamiento, es decir, la generación de señales adecuadas que actuaran sobre los motores/electroválvulas del equipo de aplicación de tratamientos.

- *Homeland sEcurity: tecnologías Para la sEguridad integRal en espacios públicos e infraestructuras (HESPERIA)*. Proyecto de investigación subvencionado por el CDTI a través del programa CENIT (Consortios Estratégicos Nacionales en Investigación Técnica) desarrollado durante los años 2008-2010 por el consorcio formado por Indra Software Labs, Unión Fenosa, Tecnobit, SAC Control, Technosafe, Visual Tools y Brainstorm Multimedia, la Universidad de Castilla-La Mancha, la Universidad de Granada, la Universidad de Extremadura, Universidad Politécnica de Madrid, la Universidad de Las Palmas de Gran Canaria, la Universidad Politécnica de Valencia, la Universidad Politécnica de Cataluña, el Centro de Automática y Robótica del CSIC y el Centro Tecnológico del País Vasco (Ikerlan). El objetivo del proyecto era el desarrollo de tecnologías que permitieran la creación de sistemas punteros de seguridad, vídeo vigilancia y control de operaciones de infraestructuras y espacios públicos. El proyecto surgió para dar respuesta a una demanda sostenida a medio y largo plazo, en particular, en países de la Unión Europea y en Estados Unidos. La gestión integrada de seguridad y control de operaciones permitía la implantación de sistemas rentables que, en ese momento, no existían en el mercado. Las tecnologías del proyecto abordaban la seguridad en dos tipos de escenarios. Por un lado, ayudaban a gestionar la seguridad y las operaciones de infraestructuras públicas especialmente sensibles, como subestaciones eléctricas, en gas, depósitos de agua o estaciones de telecomunicaciones. Por otro, incrementaban de forma sustancial los niveles de seguridad de grandes espacios públicos, como aeropuertos, estaciones de ferrocarril, puertos, centros de ciudades, zonas peatonales, centros comerciales, etc.

Los tres primeros proyectos, RHEA, GroW y PERICIA, estaban orientados al diseño y desarrollo de tecnología para Agricultura de Precisión. Permitieron la investigación en navegación autónoma basada en GPS o en el seguimiento de líneas de cultivo, desarrollo de flotas de robot autónomos para inspección y tratamiento, detección efectiva (aérea y terrestre) de malas hierbas, aplicación localizada de herbicidas mediante el control de apertura y cierre de boquillas en barras de pulverización, control mecánico y térmico de malas hierbas, pulverizadores para árboles con boquillas móviles que mejoran la penetración del tratamiento y muchas otras tecnologías que se irán presentando organizadamente a lo largo de esta memoria. Respecto al proyecto HESPERIA el trabajo realizado por el Centro de Automática y Robótica (antes Instituto de Automática Industrial) se centró en el desarrollo del Sistema de Gestión de Crisis que supervisaba la seguridad de las instalaciones y que incluía un sistema de toma decisiones que resolvía la

intrusión en una gran infraestructura decidiendo cuál era la mejor forma de distribuir los recursos (principalmente vigilantes) a partir del comportamiento presumible en el intruso. En caso de detectar una intrusión, se notificaba a los vigilantes de seguridad mediante alarmas y, teniendo en cuenta las zonas sensibles de la infraestructura y los puntos de posible huida, así como la trayectoria conocida del intruso (generada a partir de los puntos en los que previamente se le había localizado), se obtenía un plan probable de movimiento futuro del intruso que permitía determinar la mejor distribución a emplear por los vigilantes para interceptarle lo antes posible. En definitiva, se desarrollaron distintas estrategias de supervisión para alertar a los vigilantes, y se desarrollaron también estrategias de segmentación del espacio y planificación de caminos, con la finalidad de encontrar las rutas más cortas para neutralizar la intrusión. El sistema de vigilancia utilizaba alarmas para notificar a los vigilantes a cargo de la seguridad del recinto y fue la antesala de la supervisión propuesta en el Capítulo 4 de esta memoria. El sistema también utilizaba un algoritmo basado en teoría de juegos con un planificador integrado y aportó las pautas para la investigación en planificación de actuaciones en campos de cultivos que se describe en el Capítulo 3 de esta memoria.

1.3. Objetivos

El objetivo principal del presente trabajo de investigación es el desarrollo de todos los sistemas necesarios para realizar una gestión adecuada de una flota de robots autónomos aéreos y terrestres (también llamados unidades o simplemente vehículos a lo largo de la memoria), que realizan operaciones de inspección y tratamiento en entornos agrícolas.

El objetivo principal se desgrana en tres grandes objetivos con sus correspondientes sub-objetivos:

1. **Planificación:** Diseño e implementación de un Planificador que determine las actuaciones que deben realizar los vehículos terrestres de una flota para ejecutar una tarea agrícola de forma óptima.
 1. Generación de un plan de actuación que no solo contenga las trayectorias a seguir sino también las acciones necesarias para alcanzar al objetivo y que además tenga en cuenta aspectos tales como el estado del apero o la velocidad de trabajo.
 2. Caracterización de los tipos de cultivo posibles (leñosos, herbáceos, de calle ancha, de calle estrecha...) en diferentes escenarios y generalización de los procedimientos del Planificador para abarcarlos todos.
 3. Generalización del Planificador para que admita varios vehículos y con características heterogéneas (diferentes velocidades de trabajo, tanques de almacenamiento, radios de giro, consumos de combustible, etc.). La planificación debe explotar las características de cada vehículo que más benefician a la tarea global.
 4. Consideración de la variabilidad del campo, por ejemplo, la distribución de las malas hierbas, para determinar más eficientemente los planes

óptimos en aquellas tareas en las que esta información tenga un impacto cualitativo y cuantitativo.

5. Posibilidad de coberturas totales o parciales del campo en tratamientos localizados. En ocasiones los tratamientos localizados pueden realizar la selección del lugar de aplicación en tiempo real, por ejemplo, detectando las malas hierbas con cámaras incorporadas en el vehículo a medida que éste se mueve, o conocer el lugar exacto para la aplicación de antemano gracias a un mapa que indique la distribución de los rodales de malas hierbas en el cultivo. En el primer caso es necesario realizar una cobertura total del campo compaginando inspección y actuación, mientras que en el segundo la cobertura puede ser parcial, es decir, directa a las zonas con malas hierbas y por tanto sin necesidad de recorrer todo el cultivo.
 6. Análisis y estudio realista de las características de los vehículos y sus limitaciones que influyen en las trayectorias: giros limitados, tanques (de combustible, herbicida, etc.) con capacidad limitada que no permiten tratar la totalidad del campo, por lo que es necesario considerar reabastecimientos en puntos de carga y descarga, también llamados depósitos.
 7. Optimización en función de diferentes criterios: distancia, coste de los insumos agrícolas (combustible, fitosanitarios, etc.), tiempo de trabajo, etc. Los criterios se deben considerar tanto de forma independiente como combinada.
 8. Posibilidad de planificar globalmente campos complejos, esto es, irregulares y/o formados por varias parcelas no adyacentes.
 9. Validación teórica mediante la resolución de problemas reales y variados comparando los resultados con aproximaciones anteriores propuestas por otros autores. Validación real con pruebas experimentales, realizando el tratamiento con una flota de robots autónomos en campos reales.
2. **Supervisión:** Diseño e implementación de un sistema Supervisor que monitorice los robots de la flota mientras ejecutan la misión de inspección o tratamiento para garantizar que la tarea se realiza según lo planificado, ya que la incertidumbre propia de los entornos exteriores (condiciones del terreno, clima, personas, animales, etc.) puede introducir desviaciones sobre el plan inicialmente previsto.
1. Diseñar un sistema que supervise el funcionamiento de todos los elementos que integran los vehículos y que tengan relevancia en la tarea, tales como sensores, aperos, sistemas de control, etc.
 2. El sistema debe generar información clara y concisa que permita a un único operario a cargo de toda la flota entender rápidamente cómo se está desarrollando la misión y si está sucediendo algo inesperado.
 3. Diseñar el sistema para que combine de forma clara y eficiente dos de los tres aspectos principales de la supervisión: detección y diagnóstico.
 4. Además de los habituales comportamientos de detección y diagnóstico, inclusión de un comportamiento de reparación/corrección capaz de subsanar las anomalías detectadas por la monitorización, con capacidad para corregir velocidades, trayectorias o estados de los actuadores.

5. Diseñar el sistema de forma escalable de tal modo que, en caso de integrar nuevos dispositivos en los vehículos, puedan incluirse fácilmente dentro de la supervisión.
 6. Implementarlo para que sea capaz de trabajar con gran cantidad de datos en tiempo real.
 7. Evaluación del sistema Supervisor diseñado y desarrollado en campo en tareas complejas en las que intervengan varios vehículos autónomos.
3. **Gestión:** Desarrollo de un sistema Gestor global que enlace las distintas etapas involucradas en la ejecución de una misión en un escenario agrícola complejo con el fin de automatizar por completo el proceso.
1. Caracterización como un sistema modular y distribuido en el que se puedan conectar y desconectar diferentes subsistemas: planificadores de misiones de inspección/tratamiento, supervisores, módulos para procesar los datos adquiridos durante las misiones de inspección, interfaces para el operario, bases de datos, etc.
 2. Automatización de un flujo de trabajo que permita la planificación, ejecución y supervisión de una misión de inspección aérea, el procesamiento de los datos adquiridos por la inspección. Asimismo, debe coordinar la planificación, ejecución y supervisión de una misión de tratamiento terrestre a partir de los mapas generados con los datos de inspección.
 3. Diseño y desarrollo de un sistema que encaje la planificación y supervisión durante la ejecución de una misión de inspección o tratamiento. Es decir, de un módulo encargado de enviar los planes de actuación generados por el Planificador a cada uno de los vehículos de la flota involucrados en la misión, para a continuación poner en marcha el Supervisor, antes de que comience la ejecución de la misión.
 4. Diseño y desarrollo de un módulo que permita gobernar de forma básica pero suficiente el desarrollo de las misiones mediante comandos sencillos tales como pausar, reanudar o parar la misión coordinando la flota como un equipo. Se caracterizará como un sistema reactivo por interactuar con múltiples sistemas externos: cada vehículo por separado, operario, red de comunicaciones, etc.
 5. Desarrollo de un protocolo de comunicación que haga el sistema accesible desde interfaces externas para que el operario pueda acceder a los diferentes módulos que conforman el sistema.
 6. Desarrollo de las interfaces necesarias para interactuar con el sistema Gestor, que permitan visualizar con claridad la información generada por los subsistemas que lo integran: planes, averías detectadas, estado de la flota, etc.
 7. Validar el sistema mediante la ejecución completa de un escenario agrícola complejo que involucre todos los módulos necesarios para combinar satisfactoriamente una misión aérea de inspección y una misión terrestre de tratamiento.

Para validar la consecución de los objetivos anteriores, se propone alcanzar los hitos en los escenarios de prueba definidos en el proyecto RHEA. Estos escenarios tienen que ver con el tratamiento localizado de malas hierbas y con la pulverización de fitosanitarios en árboles. Los escenarios considerados corresponden a tres tipos de cultivo diferentes, a saber, de calle estrecha, de este tipo son los cultivos de cereal de invierno como el trigo, la cebada o el centeno; de calle ancha, por ejemplo los cultivos de maíz, girasol, remolacha o tomates, y, finalmente, los cultivos leñosos como los olivos, álamos, almendros, nogales, etc. La Tabla 1.2 muestra las hectáreas que en 2014 se dedicaron en Europa a alguno de los anteriores cultivos así como la producción asociada. Estos datos permiten estimar la importancia que tiene abordar el tratamiento en estos escenarios.

Tabla 1.2. Superficies y producción de varios tipos de cultivo en la Unión Europea de los 28 en 2014 según la Comisión Europea para el Desarrollo de la Agricultura (Agriculture in the European Union - Markets Statistical Information 2014, 2014)

	Calle estrecha			Calle ancha					Leñosos
	Trigo	Cebada	Total	Maíz	Girasol	Remolacha Azucarera	Tomate	Total	Olivos
Superficie ha (x1000)	23338	12668	36006	9653	4575	1512	244	15984	5000
Producción Ton (x1000)	135928	60974	196902	65461	9170	16818	14896	106345	18500

En RHEA se pusieron a punto distintas tecnologías y cada una de ellas se probó en el escenario que a priori parecía beneficiarse más. Por ejemplo, en campos de calle estrecha, donde es difícil detectar las malas hierbas mediante cámaras a bordo de los vehículos terrestres, porque debido a la densidad del cultivo y a la similitud entre malas hierbas y cultivo, se probó la conveniencia de utilizar drones en la inspección con la esperanza de que el cambio de perspectiva en la toma de imágenes permitiese detectar las zonas de rodales de malas hierbas a partir de la detección de zonas con una mayor densidad de vegetación. Por el contrario, la detección de malas hierbas en tiempo real parece factible en los cultivos de calle ancha porque es posible determinar qué zonas de la capa vegetal son líneas de cultivo y considerar como mala hierba toda la capa vegetal que se encuentra fuera de las líneas de cultivo. Asimismo, en los cultivos de calle ancha es posible controlar mecánicamente las malas hierbas, por ejemplo, arando entre las líneas de cultivo (escarda mecánica), algo que no es viable en los cultivos de calle estrecha. Teniendo en cuenta lo anterior, para los cultivos de calle estrecha RHEA optó por combinar una inspección con vehículos aéreos para determinar la distribución de las malas hierbas (mapa), con un tratamiento de pulverización localizada con vehículos terrestres. Mientras que para los cultivos de calle ancha, RHEA proponía la detección de las zonas de mala hierba en tiempo real a partir de una cámara situada en la parte frontal del tractor. En este caso el control de las malas hierbas era mecánico-térmico, con un apero formado por un cultivador que araba continuamente entre las líneas de cultivo y unos quemadores que actuaban sobre las líneas de cultivo cuando las infestaciones se eran detectadas con la cámara de a bordo. La llama generada por los quemadores tiene el efecto de reseca las plantas que, con el tiempo, acaban muriendo. Este tipo de control solo es efectivo en aquellos cultivos, como el maíz, donde el tallo de la planta crece envuelto por capas de hojas, capaces de proteger las plantas de las pequeñas llamas de los

quemadores. Con el tiempo las hojas de las capas más exteriores se resecan y se caen, pero el tallo sigue intacto y la planta sobrevive, en cambio las plantas de mala hierba, que no suelen tener esta protección, mueren.

El tercer escenario de RHEA aborda los cultivos leñosos, en concreto en aquellos escenarios en los que es necesario aplicar alguna clase de fitosanitario en la copa de los árboles bien de modo preventivo o para paliar algún tipo de plaga (hongos, bacterias, insectos, etc.). En estos casos lo habitual es aplicar el tratamiento a todo el cultivo. Para este escenario RHEA utilizó un pulverizador de árboles que se accionaba cuando se detectaba la copa de un árbol y además las boquillas en los extremos se movían para adaptarse al máximo al tamaño del árbol.

1.4. Publicaciones

La investigación llevada a cabo durante el desarrollo de la presente tesis ha dado lugar a numerosas contribuciones en los campos de planificación y supervisión de flotas de robots, optimización combinatoria e incluso detección de malas hierbas, siendo de gran relevancia e innovación en estos contextos. Las publicaciones científicas que atestiguan estas aportaciones se listan a continuación.

1.4.1. Revistas

Las publicaciones en revistas se han ordenado por importancia.

Publicación 1. Conesa-Muñoz, J., Gonzalez-de-Soto, M., Gonzalez-de-Santos, P., & Ribeiro, A. (2015). Distributed multi-level supervision to effectively monitor the operations of a fleet of autonomous vehicles in agricultural tasks. *Sensors*, 15(3), 5402-5428.

En este artículo se propone una arquitectura de supervisión multi-nivel y distribuida para abordar uno de los grandes temas de la tesis: la supervisión de los trabajos agrícolas llevados a cabo por flotas de vehículos autónomos (explicado detalladamente en el Capítulo 4 de la presente memoria). La arquitectura divide la supervisión en diferentes módulos o bloques de monitorización que pueden estar distribuidos, que interactúan entre sí mediante alarmas y que permiten construir y organizar mejor la detección, el diagnóstico y la reparación de los fallos.

A partir de la arquitectura propuesta (sección 4.1) se construyó un sistema supervisor (sección 4.3) de alto nivel capaz de vigilar que los vehículos autónomos de una flota ejecuten, según lo previsto, las tareas agrícolas encomendadas.

El sistema se verificó con una flota real de tractores autónomos (flota de RHEA, ver página web <http://www.rhea-project.eu/>) ejecutando varias tareas agrícolas y se comprobó que detectaba adecuadamente problemas tales como desviaciones en la velocidad de trabajo, desviaciones en las trayectorias o posibles riesgos de colisión entre dos o más vehículos. En este último caso, incluso fue capaz de neutralizar las colisiones actuando directamente sobre los robots y sin necesidad de interrumpir la misión. Los resultados se muestran en la sección 4.4.

Publicación 2. Conesa-Muñoz, J., & Ribeiro, A. (2012). Consolidation of a WSN and minimax method to rapidly neutralise intruders in strategic installations. *Sensors*, 12(3), 3281-3301.

En esta investigación se trabajó en combinar una red inalámbrica de sensores de movimiento y un algoritmo computacional de tipo Minimax, propio de Inteligencia Artificial, para detectar y neutralizar intrusiones físicas en infraestructuras de vital importancia. Se desarrolló un sistema de decisión que era capaz de determinar en tiempo real qué estrategia seguir para neutralizar la intrusión. Una red de sensores de movimiento informaba al sistema de las zonas en las que detectaba actividad de personal sin identificar, y el sistema infería las trayectorias de los intrusos por las zonas en las que se detectaba actividad y escogía la mejor estrategia para acabar con la intrusión planteando la situación como un juego que debía ganar. Para escoger el mejor movimiento con el que encauzar el juego hacia la victoria, se utilizó un algoritmo Minimax que interpretaba a los intrusos como un jugador que buscaba maximizar el daño y al sistema de seguridad como el oponente que debía minimizarlo. Para medir los posibles daños potenciales el algoritmo evaluaba las distancias entre los intrusos, los guardias, las salidas y los objetivos más sensibles. Entre las actuaciones posibles propuestas por el sistema para neutralizar la amenaza se encontraban activar alarmas, cerrar puertas o dirigir a los guardias hacia determinadas zonas. Fue esencial trabajar en técnicas de planificación de rutas en entornos con obstáculos, tales como las triangulaciones de Delaunay y el algoritmo A*. Las estrategias de planificación que se desarrollaron se utilizaron más tarde como primera aproximación al problema de la planificación agrícola, aunque posteriormente se obtuvieron mejores resultados empleando estrategias meta-heurísticas por su mayor idoneidad tal como se explica en la memoria en la sección 2.2. Asimismo, la idea de supervisión ya aparece en este trabajo, en el que en función de la monitorización continua de las instalaciones, se generan alarmas y se buscan soluciones para paliar los problemas. La estructura de supervisión así como la gestión conceptual propuesta en este trabajo para gestionar una crisis es el embrión de la estructura de supervisión y gestión de misiones propuesta en esta tesis para abordar con éxito la realización de una tarea agrícola de cierta complejidad.

Publicación 3. Ribeiro, Á., Sainz-Costa, N., Herrera, P. J., Conesa-Muñoz, J., Fernández-Quintanilla, C., Andujar, D., & Dorado, J. (2013). Optimización evolutiva de un método de discriminación visual entre mono y dicotiledóneas: resultados iniciales. *Interempresas*, 114(1016), 114-122.

En este trabajo se propone un método basado en algoritmos evolutivos para encontrar el procesamiento óptimo a realizar sobre una imagen binaria, resultado de un proceso de segmentación de la capa vegetal en una imagen RGB, para aislar la cubierta de dicotiledóneas y con ello poder calcular los tantos por ciento en la imagen de cubierta de monocotiledóneas y de dicotiledóneas.

Se colaboró en el sistema de discriminación que se desarrolló basado en algoritmos genéticos, aprovechando la experiencia adquirida en este tipo de técnicas, ya que en la tesis se proponen distintas aproximaciones basadas en técnicas evolutivas (secciones 3.2.8 y 3.2.9). En concreto, se definió la representación de la solución, los operadores de cruce y mutación, y la función de fitness.

1.4.2. Capítulos de libros

Los capítulos en libros están ordenados por importancia.

Publicación 4. Burgos-Artizzu, X. P., Conesa-Muñoz, J., & Ribeiro, A. (2010). Razonamiento basado en casos. En G. Pajares, *Aprendizaje automático: un enfoque práctico* (págs. 309-327). RA-MA Editorial. Madrid.

El libro trata sobre aplicaciones prácticas actuales de las técnicas más conocidas en aprendizaje automático: redes neuronales, máquinas de vectores soporte, redes bayesianas, árboles de decisión, razonamiento basado en casos, etc. Se colaboró en el capítulo dedicado al razonamiento basado en casos (CBR) aportando un caso práctico y real de CBR desarrollado para resolver situaciones de intrusión en infraestructuras sensibles como aeropuertos, edificios gubernamentales, etc. En el trabajo se usan técnicas de planificación para medir las distancias entre diferentes elementos cruciales: intrusos, guardias, salidas, etc. Estas técnicas (triangulaciones de Delaunay y algoritmo A*) también se probaron para abordar el problema de la planificación agrícola aunque más tarde se descartaron al encontrar mejores estrategias como se explica en la sección 2.2 de la presente memoria.

Publicación 5. Conesa-Muñoz, J., & Ribeiro, A. (2012). An evolutionary approach to optimally distributing a fleet of robots for weed control in arable crops based on economical costs. En *ICT for agriculture, rural development and environment: Where we are? Where we will go?* (págs. 280-299). Praga.

El libro está dedicado a las principales tendencias futuras de la agricultura. Se extendió el trabajo publicado en el congreso EFITA11 (**Publicación 21**) estudiando el problema de la planificación de rutas en campos de cultivo desde un punto de vista económico (costes explicados en la sección 3.2.3.1). Esto es, se optimizaron las rutas empleando un algoritmo evolutivo cuyo objetivo principal era reducir al máximo los costes económicos de la tarea (debidos al combustible utilizado, tiempo requerido y herbicida aplicado). La aproximación propuesta está pensada para ser aplicada dentro de un esquema de gestión de Agricultura de Precisión en el que varios vehículos realizan el tratamiento simultáneamente, de ahí la novedad (recogido en la formulación planteada en la sección 3.1) y que el trabajo constituya un capítulo de este libro.

1.4.3. Congresos

En el caso de los congresos, las publicaciones están ordenadas por orden cronológico lo que permite ver la evolución del trabajo desarrollado a lo largo de estos años y presentado en la memoria de tesis.

Publicación 6. Conesa-Muñoz, J., Valente, J., Del Cerro, J., Barrientos, A., & Ribeiro, A. (2015). Integrating autonomous aerial scouting with autonomous ground actuation to reduce chemical pollution on crop soil. En *Proceedings of ROBOT2015: Second Iberian Robotics Conference*. Lisboa: Springer International Publishing.

En el trabajo se propone la integración de misiones aéreas de inspección con misiones terrestres de tratamiento para abordar problemas medioambientales en los que, debido a la gran cantidad de superficie a cubrir, es más eficiente dividir en dos etapas la resolución del problema. Se desarrollaron los sistemas necesarios para la etapa de tratamiento terrestre (planificador, supervisor y controlador de flota terrestre) y para la integración con los demás sistemas. En este artículo se presenta el sistema gestor propuesto en la tesis (en el Capítulo 4.4) y que permite automatizar por completo las misiones agrícolas.

Publicación 7. Ribeiro, A., Fernandez-Quintanilla, C., Dorado, J., López-Granados, F., Manuel Peña, J., Rabatel, G., Pérez-Ruiz, M., **Conesa-Muñoz, J.**, & Gonzalez-de-Santos, P. (2015). A fleet of aerial and ground robots: a scalable approach for autonomous site-specific herbicide application. En *Proceedings of 10th European Conference on Precision Agriculture*, (págs. 167-173).

En el artículo presentado se propone la combinación de una misión de inspección aérea con una de tratamiento terrestre para la aplicación localizada de herbicida. Se colaboró realizando la parte de planificación y supervisión de la flota terrestre propuesta en el trabajo, que corresponden con la planificación y supervisión expuestas en la tesis (en los Capítulos 3 y 4).

Publicación 8. Conesa-Munoz, J., Bengochea-Guevara, J. M., Andujar, D., & Ribeiro, A. (2015, April). Efficient Distribution of a Fleet of Heterogeneous Vehicles in Agriculture: A Practical Approach to Multi-path Planning. En *Autonomous Robot Systems and Competitions (ICARSC), 2015 IEEE International Conference on* (págs. 56-61). IEEE.

En el artículo se recopilan todas las propuestas novedosas de planificación planteadas en trabajos previos (**Publicación 12**, **Publicación 13**, **Publicación 16**, **Publicación 18**, **Publicación 20**, **Publicación 21**) y se propone un planificador muy general que considera flotas heterogéneas, variabilidad en el cultivo, operaciones de reabastecimiento y diferentes criterios de optimización; cubriendo, en definitiva, una gran cantidad de situaciones agrícolas reales (corresponde con el Planificador planteado en el Capítulo 3).

Publicación 9. Conesa-Muñoz, J., Emmi, L., Gonzalez-de-Soto, M., & Ribeiro, A. (2014). Effective supervision of a tractor fleet. En *Robotics and associated High-technologies and Equipment for Agriculture and forestry (RHEA 2014), 2nd International Conference on* (págs. 579-587). Madrid.

En el trabajo se propone una supervisión eficiente para una flota de tractores combinando una monitorización a bajo nivel, encargada de vigilar a cada uno de los tractores por separado, y otra a mayor nivel, encargada de supervisar a la flota en su conjunto. Esta supervisión corresponde con la propuesta en la tesis en el Capítulo 4. La publicación **Publicación 1**, referenciada al inicio de la sección 1.4.1, es una ampliación que recoge más detalles y experimentos de validación de la aproximación a la supervisión presentada en este congreso.

Publicación 10. Bengochea-Guevara, J. M., **Conesa-Muñoz, J.**, & Ribeiro, Á. (2014, January). Generating Autonomous Behaviour for a Crop Inspection Robot. En *ROBOT2013: First Iberian Robotics Conference* (pp. 481-493). Springer International Publishing.

En este trabajo se desarrolló un controlador para un robot autónomo de inspección por un cultivo de pequeña envergadura. Se colaboró ayudando en la puesta en marcha del robot y en la navegación autónoma que incluía una etapa de planificación y un módulo de supervisión basados en parte de las ideas propuesta en la tesis (planificación y supervisión descritas en los Capítulos 3 y 4, respectivamente).

Publicación 11. Ribeiro, A., Sainz-Costa, N., Herrera, P.J., **Conesa-Muñoz, J.**, Fernández-Quintanilla, C., Andujar, D., & Dorado, J. (2013). Learning the best segmentation process for discriminating accurately between monocots and dicots. En *Precision Agriculture (ECPA 2013), 9th European Conference on*. Book of posters.

En este trabajo se propuso un Algoritmo Genético para discriminar visualmente entre monocotiledóneas y dicotiledóneas. Se colaboró en el desarrollo del método evolutivo, aprovechando la experiencia en este tipo de técnicas de optimización (utilizadas en la tesis en las secciones 3.2.8, 3.3.3 y 3.4.2), trabajando en la representación de la solución y en los operadores de cruce y mutación.

Publicación 12. **Conesa-Muñoz, J.**, Bengochea-Guevara, J. M., & Ribeiro, A. (2013). Path planning to minimise distances and recharging instances for a small fleet of vehicles in an arable field. En *Proceedings of Precision agriculture'13* (págs. 373-380). Wageningen Academic Publishers.

En este trabajo se estudió cómo afecta en la planificación la distribución de pequeñas estaciones de recarga en las cabeceras de un cultivo, necesarias para abastecer pequeñas plataformas móviles con autonomía limitada. El trabajo se centró en determinar, en función del campo, el número óptimo de robots que debería tener la flota y el número de estaciones necesarias, así como su ubicación, para dar servicio adecuadamente al conjunto de robots.

Este trabajo permitió estudiar el problema de las recargas/reabastecimientos de los vehículos en las planificaciones agrícolas, lo que contribuyó a generalizar la planificación propuesta en la tesis, abarcando escenarios más realistas en los que los vehículos no tienen capacidad de tanque (de herbicida, combustible, etc.) suficiente para cubrir todo el campo (Secciones 3.1, 3.2.2 y 3.4.3).

Publicación 13. **Conesa-Muñoz, J.**, Ribeiro, A., Andujar, D., Fernandez-Quintanilla, C., & Dorado, J. (2012). Exploring the effect of turning maneuvers on the multi-path planning of a robot fleet in agricultural tasks. En *Robotics and associated High-technologies and Equipment for Agriculture (RHEA 2012), 1st International Conference on*, (págs. 339-344). Pisa.

En este trabajo de investigación se analizó el efecto que tienen las maniobras de giro en el coste de las trayectorias sobre el cultivo y la forma, por tanto, cómo éstas

afectan a la planificación agrícola. Hasta este trabajo no se habían considerado las maniobras reales de giro de los tractores convencionales en las planificaciones (maniobras detalladas en la sección 3.2.2.1 de la memoria).

Publicación 14. Fernandez-Quintanilla, C., Dorado, J., **Conesa-Muñoz, J.**, & Ribeiro, A. (2012). WEEDEX – a weed expert system for robot fleet in agriculture. En *Robotics and associated High-technologies and Equipment for Agriculture (RHEA 2012), 1st International Conference on*, (págs. 363-368). Pisa.

En este trabajo se diseñó un sistema experto con el que tomar decisiones adecuadas para controlar las malas hierbas. Estas decisiones son parte de la planificación del trabajo de un conjunto de robots autónomos que transportan los elementos de actuación en el campo (barra de aplicación de herbicida, aforo de control mecánico y térmico o equipo de pulverización de árboles). Se colaboró aportando las etapas de planificación y supervisión del tratamiento con la flota de robots, con técnicas similares a las que se describen en la memoria de la tesis en los Capítulos 3 y 4.

Publicación 15. Bengochea-Guevara, J. M., **Conesa-Muñoz, J.**, & Ribeiro, Á. (2012). Developing a small autonomous robot for crop inspection. En *Information Technology, Automation and Precision Farming. International Conference of Agricultural Engineering-CIGR-AgEng 2012: Agriculture and Engineering for a Healthier Life, Valencia, Spain, 8-12 July 2012* (págs. C-2171). CIGR-EurAgEng.

Se colaboró en el desarrollo del controlador interno que permitía gobernar el robot en modo tele-operado y en modo completamente autónomo siguiendo un plan de ruta previamente establecido. Por tanto, se aportaron los mecanismos de planificación y supervisión que se utilizaron en este trabajo y que se encuentran descritos en la tesis en los Capítulos 3 y 4.

Publicación 16. **Conesa-Muñoz, J.**, Ribeiro, A., Andujar, D., Fernandez-Quintanilla, C., & Dorado, J. (2012, June). Multi-path planning based on a NSGA-II for a fleet of robots to work on agricultural tasks. En *Evolutionary Computation (CEC), 2012 IEEE Congress on* (págs. 2236-2243). IEEE.

En este trabajo se evalúa un método de optimización multi-objetivo basado en el algoritmo NSGA-II, como técnica para obtener las trayectorias óptimas en campos de cultivo. Mediante este planteamiento se optimizaron las trayectorias para minimizar simultáneamente la distancia recorrida, el tiempo requerido para realizar la tarea y coste económico en términos de insumos utilizados. Este trabajo contribuyó a la tesis aportando una nueva técnica de planificación, en este caso, multi-objetivo (secciones 3.2.9, 3.3.4 y 3.4.2).

Publicación 17. **Conesa-Muñoz, J.**, Gottschalk, R., Burgos-Artizzu, X. P., & Ribeiro, A. (2011). Método de detección visual de líneas de cultivo para el control en dirección de vehículos agrícolas. En *Conferencia de la Asociación Española para la Inteligencia Artificial (CAEPIA)*.

Se diseñó un método de detección visual de líneas de cultivo para pequeños vehículos agrícolas y se comparó con otros métodos ya existentes. Este método puede integrarse con la planificación y supervisión expuestas en la tesis (en los Capítulos 3 y 4) y cerrar así la navegación autónoma de un pequeño vehículo para que cubra completamente un cultivo.

Publicación 18. Conesa-Munoz, J., Ribeiro, A., & Pajares, G. (2011). A multi-path planning approach based on a genetic algorithm for a robot fleet working in arable crops. En *Proceedings of the Portuguese Conference on Artificial Intelligence (EPIA'11)*. (págs. 167-181).

En este trabajo se implementó la planificación agrícola mediante un Algoritmo Genético para encontrar las rutas óptimas en un campo de cultivo para una flota de vehículos. Este trabajo es una extensión de la **Publicación 21**, en el que se considera también el criterio del tiempo a la hora de optimizar las rutas. Estas contribuciones se plasman en la tesis en la planificación en las secciones 3.2.3, 3.2.8 y 3.4.3.

Publicación 19. Fernandez-Quintanilla, C., Dorado, J., San Martin, C., Conesa-Muñoz, J., & Ribeiro, A., (2011). A five-step guide for planning a robotic site-specific weed management program for winter wheat. En *Proceedings of the First International Workshop on Robotics and Associated High-Technologies and Equipment for Agriculture*, (págs. 3-11). Montpellier.

En el trabajo se propone un método en varios pasos para eliminar malas hierbas. Se colaboró aportando los pasos de planificación y supervisión. Se trata de un trabajo de investigación inicial sobre planificación y supervisión que evolucionó posteriormente hasta alcanzar el nivel de madurez de las estrategias y sistemas que se explican en detalle en la tesis (en los Capítulos 3 y 4).

Publicación 20. Ribeiro, A., & Conesa-Muñoz, J., (2011). Path-planning of a Robot Fleet Working in Arable Crops. First experiments and results. En *Proceedings of the First International Workshop on Robotics and Associated High-Technologies and Equipment for Agriculture*, (págs. 117-128). Montpellier.

En el trabajo se plantea el problema de planificación agrícola y se aporta una primera solución usando por primera vez el algoritmo Recocido Simulado (descrito en las secciones 3.2.7, 3.3.2 y 3.4). El trabajo descrito en este artículo es una de las versiones iniciales de las meta-heurísticas propuestas para la planificación y explicadas en detalle en la memoria (en la sección 3.2).

Publicación 21. Conesa-Munoz, J., & Ribeiro, A. (2011, July). An evolutionary approach to obtain the optimal distribution of a robot fleet for weed control in arable crops. En *Proceedings of the 8th European Federation for Information Technology in Agriculture, Food and the Environment/World Congress on Computers in Agriculture. Prague, Czech Republic* (págs. 141-155).

En el artículo se utiliza un Algoritmo Genético para estudiar una versión simplificada del problema de planificación agrícola mediante una flota de robots. Este método es una de las versiones iniciales que ha ido madurando a lo largo de estos años para convertirse en la propuesta que se detalla en la memoria (secciones 3.2.8, 3.3.3 y 3.4.2).

Publicación 22. Conesa-Muñoz, J., & Ribeiro, A. (2010). Planificación óptima de una flota de robots en operaciones agrícolas. En *Robocity2030: Robots de exteriores* (págs. 183-197). Madrid.

En este artículo se habla sobre planificación de robots para operaciones agrícolas y explica el protocolo general que hay que seguir para obtener trayectorias realistas en entornos de trabajo agrícola. Respecto a su relación con la tesis, esta propuesta es una primera versión de la planificación explicada en el Capítulo 3.

Publicación 23. Ribeiro, A., Gottschalk, R., Burgos-Artizzu, X. P., & Conesa-Muñoz, J. M. (2010) Reconocimiento visual en tiempo real de líneas de cultivo y su aplicación para la navegación de un robot móvil de pequeña envergadura. En *Robocity2030: Visión en robótica* (págs. 183-198). Madrid.

En este trabajo se explica un método de reconocimiento visual en tiempo real de líneas de cultivo para que un robot navegue de forma autónoma por el interior de un cultivo siguiendo las líneas. Las líneas de cultivo se detectan a partir de las imágenes suministradas por la cámara de a bordo del robot. Así una vez que el vehículo autónomo recibe el plan de ruta, obtenido a partir de las estrategias de planificación propuestas en la presente memoria en la sección 3.2, el vehículo tiene que ir desde un punto de entrada al campo hacia un punto de salida sin pisar las líneas de cultivo. Esta forma de navegación tiene interés y es esencial en los casos en los que el cultivo no es capaz de recuperarse una vez que se pisa con las ruedas del tractor (maíz, girasol, etc.).

Publicación 24. Conesa, J., & Ribeiro, A. (2009, May). Performing the Retrieve Step in a Case-based Reasoning System for Decision Making in Intrusion Scenarios. En *Proceedings of the Artificial Intelligence and Decision Support Systems Conference (ICEIS 2009)*. *ICEIS* (2) (págs. 343-346).

Se desarrolló un sistema de razonamiento basado en casos para detectar por similitud qué estrategia de neutralización aplicar en una crisis de intrusión en una infraestructura cerrada. En el trabajo se usan técnicas de planificación para medir las distancias entre diferentes elementos cruciales: intrusos, guardias, salidas, etc. Estas técnicas (triangulaciones de Delaunay y algoritmo A*) se utilizaron también en la resolución del problema de la planificación en el ámbito agrícola, aunque posteriormente se emplearon técnicas más adecuadas (meta-heurísticas detalladas en la sección 3.2) como se explica en la tesis (sección 2.2). Aun así este trabajo fue un primer contacto con los temas de planificación y aportó una gran base teórica que ha sido esencial a la hora de abordar adecuadamente el problema de la planificación en el entorno agrícola descrito en el Capítulo 3 de la tesis.

1.5. Estructura de la tesis

La presente memoria está dividida en los capítulos que se describen a continuación.

El capítulo 1 o de *introducción* describe el problema a investigar de forma general, así como la motivación, los objetivos y las contribuciones de la presente tesis.

El capítulo 2, o del *estado del arte*, revisa la situación actual y avances las diferentes temáticas que se abordan para enfrentarse a los objetivos planteados en este trabajo de investigación.

El capítulo 3, o de *planificación*, aborda en profundidad el problema de la planificación para vehículos agrícolas dotados de aperos que deben ejecutar algún tipo de tarea en un campo agrícola. Se plantea el problema de una forma muy general, esto es, considerando flotas con varios vehículos con distintas características y limitaciones, variabilidad en el cultivo, múltiples criterios de optimización, etc. El problema se expresa como un problema de optimización combinatoria y se resuelve probando diferentes métodos meta-heurísticos. Asimismo, se propone un nuevo operador para construir las nuevas soluciones durante el proceso de exploración del espacio de búsqueda. La aproximación propuesta se valida con la resolución de un conjunto de ejemplos ilustrativos de los que es posible conocer las características de la solución óptima a priori, además de comparar los resultados del método con otros que aparecen descritos en la literatura.

El capítulo 4, o de *supervisión*, se centra en diseñar e implementar un sistema de supervisión que monitoriza la actividad de los vehículos y comprueba que las unidades de la flota están llevando a cabo la tarea según lo planificado. Para ello, se diseña una arquitectura de supervisión distribuida y multinivel, capaz de generar alarmas, que permite separar fácilmente la supervisión en bloques simples y construir incrementalmente sistemas de monitorización cada vez más complejos. El sistema se diseña e implementa para incluir las tres funcionalidades principales de la supervisión, a saber, la detección, el diagnóstico y la reparación. La aproximación propuesta se valida ejecutando misiones reales con varios vehículos y comprobando que el sistema desarrollado es capaz de detectar anomalías tales como desviaciones en la velocidad de trabajo o el riesgo de colisión entre dos o más vehículos. En algunos casos, como este último, el sistema es capaz de paliar el riesgo de colisión actuando directamente sobre los propios vehículos.

El capítulo 5, o de *gestión de misiones*, se dedica al diseño e implementación de un sistema global para automatizar totalmente las tareas complejas en escenarios agrícolas que se pueden beneficiar de la combinación de una misión aérea de inspección con una misión de intervención realizada en tierra con vehículos terrestres. Para ello, este sistema permite enlazar planificadores y supervisores, como los de los capítulos 3 y 4, tanto aéreos como terrestres, además de otros sistemas como aquellos que permiten cartografiar los rodales de malas hierbas, a partir de la composición de las fotos tomadas por los vehículos aéreos en una sola imagen (proceso de mosaicado) y la posterior discriminación de la capa vegetal de maleza (generación del mapa de distribución de mala hierba). El sistema de gestión desarrollado se valida realizando una misión agrícola compleja, en concreto un tratamiento con pulverización de rodales de malas hierbas en un cultivo de trigo, de forma totalmente automatizada, integrando una parte aérea de inspección y una parte terrestre de tratamiento.

El capítulo 6, o de *conclusiones y trabajo futuro*, expone las principales conclusiones del trabajo de investigación descrito a lo largo de esta memoria a la vez que resume las líneas de investigación abiertas y que pueden abordarse en un futuro.

2 ESTADO DEL ARTE

2.1. Automatización agrícola

La productividad agrícola se ha incrementado notablemente en los últimos años gracias a la automatización (Edan, 2009). Tanto es así, que en la actualidad un agricultor en Estados Unidos produce lo suficiente para alimentar a 130 personas, mientras que a principios del siglo XX solo alcanzaba de 2 a 5 personas (Burgos Artizzu, 2009). No obstante, estas cifras todavía tienen que mejorar significativamente, ya que la población mundial no para de aumentar a la par que cada vez queda menos tierra cultivable y se dispone de menos agua. En concreto, según informes de la Global Harvest Initiative, se estima que la productividad actual tiene que duplicarse para que la agricultura pueda satisfacer las demandas (comida, pienso, fibra, biocombustible, etc.) de los 9.500 millones de personas que se calculan para el año 2050 (Global Harvest Initiative, 2014).

Dentro de la automatización agrícola, uno de los factores que más ha contribuido hasta ahora a aumentar la productividad ha sido la mecanización de la maquinaria (Reid, 2011), buena prueba de ello es la gran cantidad de avances destinados en los últimos años a automatizar cada vez más los vehículos de trabajo, entre los que destacan los progresos en la navegación semi-autónoma y en la robotización de los aperos. El otro factor que ha contribuido a la mejora en la productividad ha sido la extracción del conocimiento del cultivo y su aplicación para realizar los trabajos agrícolas de un modo más eficiente.

Respecto a la navegación autónoma, varias son las investigaciones llevadas a cabo a lo largo de los últimos años (Mousazadeh, 2013) y (Li, Imou, Wakabayashi, & Yokoyama, 2009), entre las que se pueden encontrar sistemas de navegación basados en GNSS, visión y láseres, siendo los sistemas basados en GNSS, en particular en GPS, los más utilizados. Grandes compañías como John Deere o Case New Holland ya incorporan esta tecnología en sus vehículos (John Deere Guidance Systems, 2015) (Sistemas agrícolas avanzados CIH, 2015), de modo que estos pueden moverse autónomamente dentro del cultivo aunque los cambios de dirección en cabecera siguen siendo manuales. El auge de este tipo de sistemas de navegación tiene su explicación en el gran desarrollo que han sufrido los sistemas GNSS en los últimos años y que ha permitido disponer en el mercado de receptores capaces de suministrar la señal de localización con un error por debajo del centímetro a frecuencias relativamente altas, del orden de 20 Hz. En particular estos sistemas GNSS incorporan una técnica de mejora de la precisión conocida como RTK (Real Time Kinematic), que utiliza las medidas de la fase de la onda portadora de la

señal. Con la técnica RTK al menos una estación base de referencia (fija), provista de un receptor GNSS, tiene que suministrar correcciones instantáneas, retransmitiendo la fase de la onda portadora de la señal que recibe de los satélites. El receptor GNSS (dispositivo móvil) recibirá la corrección y comparará la fase de la señal recibida de los satélites con la fase de la señal de la estación de referencia, calculando su posición relativa con precisión milimétrica al mismo tiempo que recalcula sus coordenadas globales (latitud, longitud). Otra opción para mejorar la precisión de la localización suministrada por el GNSS es tener en cuenta las restricciones cinemáticas del tractor, como se propone en (Gomez-Gil, Alonso-García, Gómez-Gil, & Stombaugh, 2011).

Dos son las principales desventajas de los sistemas de navegación basados en GNSS: 1) la posible pérdida de señal por interferencias electromagnéticas o por la frondosidad de los árboles en cultivos leñosos, y 2) el posible daño al cultivo (pisar algunas plantas) al realizar una navegación basada solo en la posición sin tener en cuenta el trazado real del cultivo, ya que esta información es difícil determinar con gran precisión en cultivos extensivos, como por ejemplo los de maíz.

Como alternativa, los sistemas de navegación basados en visión se guían tomando como referencia el cultivo. Para ello detectan las plantas, analizan la estructura en la que están dispuestas (habitualmente en líneas paralelas) y su posición, utilizando finalmente esta información para guiar la navegación de forma que no se dañe ninguna planta de cultivo. La mayoría de estos sistemas son dependientes de las condiciones de iluminación y resuelven con mayor o menor fortuna las irregularidades en el cultivo como los fallos en la siembra, que hacen desaparecer temporalmente las líneas, y los grandes rodales de malas hierbas, que enmascaran las líneas dificultando su detección. Estos sistemas se han empleado tanto en grandes plataformas (Benson, Reid, & Zhang, 2003), (Han, Zhang, Ni, & Reid), (Gottschalk, Burgos-Artizzu, Ribeiro, Pajares, & Miralles, 2009) como en pequeños vehículos de inspección (Bengochea-Guevara, Conesa-Muñoz, & Ribeiro, 2012).

Por otro lado, los sistemas basados en láser necesitan reflectores que actúen como puntos de referencia en el campo. Se obtiene el tiempo que tarda en detectarse el haz de luz y se calcula la posición del vehículo mediante triangulación. Su ventaja respecto a las otras propuestas es que son robustos frente a las condiciones meteorológicas y a las interferencias magnéticas (Li, Imou, Wakabayashi, & Yokoyama, 2009). Sin embargo, son muy dependientes de las referencias y funcionan muy mal en terrenos abruptos y/o con obstáculos, porque introducen mucho ruido en las medidas del láser. Con frecuencia se utilizan en frutales porque el follaje bloquea la recepción de señales de los GNSS (Barawid, Mizushima, Ishii, & Noguchi, 2007), (Chateau, Debain, Collange, Trassoudaine, & Alizon, 2000), (Tsubota, Noguchi, & Mizushima, 2004).

Los aperos también se han combinado, en los últimos años, con todo tipo de sensores para hacerlos más versátiles y eficientes. Por ejemplo, ya existen barras de pulverización que incorporan receptores GNSS y que son capaces de pulverizar con una precisión de centímetros (Perez-Ruiz, Carballido, Agüera, & Rodríguez-Lizana, 2013). También existen aperos que trabajan combinados con cámaras y procesos, capaces de detectar con exactitud la posición de las plantas. Por ejemplo, en (Raffaelli, Martelloni, Frasconi, Fontanelli, & Peruzzi, 2013) se usa una herramienta mecánico-térmica en cultivos de calle ancha que ara continuamente entre líneas y quema las malas hierbas de la línea de cultivo activando la llama sólo cuando se detecta una zona de mala hierbas entre líneas. Otro ejemplo se muestra en (Tillett, Hague, Grundy, & Dedousis, 2008), dónde se presenta un apero para introducir pequeñas azadas en las propias líneas de

cultivo y trabajar cada planta individualmente; la visión se utiliza en este caso para detectar la planta de cultivo

Por último, la extracción del conocimiento del cultivo también ha crecido en gran medida gracias a los numerosos avances en los sistemas de percepción y en los propios sensores (Mulla, 2013). Mientras que antes era necesario un equipo con muchas personas para muestrear un campo en un tiempo razonable, ahora es posible inspeccionarlo mucho más rápido con una flota de pequeños robots aéreos (Barrientos, y otros, 2011) o con una plataforma robótica terrestre (Bak & Jakobsen, 2004). El objetivo de todos estos trabajos es generar mapas de variabilidad del cultivo, por ejemplo, de infestación de malas hierbas detectando la posición y densidad de los rodales mediante cámaras instaladas en los propios vehículos de tratamiento (Burgos-Artizzu, Ribeiro, Guijarro, & Pajares, 2011), (Montalvo, Guerrero, Romeo, Emmi, Guijarro, & Pajares, 2013), (Guerrero, y otros, 2013) o incluso instaladas en vehículos aéreos no tripulados (Peña-Barragán, Torres-Sánchez, de Castro, Kelly, & López-Granados, 2013). Estos mapas son la base de los tratamientos localizados, en otras palabras la combinación de la información contenida en estos mapas con aperos de precisión están posibilitando la aplicación localizada de agroquímicos (Weisz, Fleischer, & Smilowitz, 1996), abonos (Dobermann, y otros, 2002) o herbicidas (Gerhards, Sökefeld, Schulze-Lohne, Mortensen, & Kühbauch, 1997).

2.2. Planificación de la tarea

Dentro de las líneas de investigación, la navegación autónoma tiene un importante interés en el entorno agrícola, ya que muchas tareas requieren recorridos que garanticen la cobertura total del cultivo sin solapamientos. En efecto, en el mercado existen muchos sistemas comerciales relacionados con esta cuestión (Topcon Precision Agriculture, 2015), (Trimble GNSS Systems, 2015), (John Deere Guidance Systems, 2015), (Sistemas agrícolas avanzados CIH, 2015) cuya finalidad, por el momento, es guiar al operario. Estos sistemas permiten fijar una ruta, ya sea manualmente o a partir de algún patrón de recorrido (Topcon System 350, 2015), y después controlar que la ruta marcada se sigue con exactitud, actuando sobre el sistema de dirección del vehículo para corregir las desviaciones. Estos sistemas son muy útiles para evitar solapamientos, y por ende reducir distancias, combustible, tiempo y compactación, lo que es especialmente difícil en campos muy grandes donde el operario puede fácilmente perder la cuenta de la superficie ya recorrida y repetir o saltarse algunas zonas. En algunos casos, estos sistemas pueden supervisar otros detalles de la tarea que se está ejecutando, por ejemplo, si el caudal de herbicida es el correcto (Topcon System 350, 2015).

Otro aspecto importante que requiere atención es determinar la ruta a seguir para realizar el trabajo de forma óptima, considerando las características del campo, los requisitos de la tarea y las características del vehículo.

Planificar las rutas a seguir es un aspecto crucial, no ya solo en aplicaciones agrícolas y/o con vehículos, sino en todo tipo de problemas, porque, por lo general, condicionan las actuaciones posteriores. Por ejemplo, en diseño de placas, el trazado de las conexiones puede condicionar donde se sitúan los componentes electrónicos (Cong, Fang, & Khoo, 2000); en neurocirugía la posición de un brazo robótico puede determinar la distribución de una dosis de radiación (Tombropoulos, Adlerb, & Latombec, 1999) o, en seguridad de grandes recintos, la distancia entre determinados elementos clave (vigilantes, intrusos, salidas, objetivos, etc.) es crucial para determinar la actuación más

apropiada, por ejemplo, para neutralizar una intrusión (Conesa-Muñoz & Ribeiro, 2012). En este último trabajo se segmenta el espacio en una triangulación de Delaunay con restricciones (Demyen & Buro, 2006), esto es, en celdas de forma triangular cuyas aristas pueden estar restringidas o no (permiten o no el paso entre los espacios adyacentes), y se emplea una adaptación del algoritmo A* para nodos con dimensión y forma triangular, conocida como TA* (Kallmann, Bieri, & Thalmann, 2004), para encontrar el camino más corto entre dos puntos cualesquiera.

En el caso de la agricultura, las trayectorias a seguir son un aspecto importante de la tarea a realizar. Tradicionalmente los agricultores han trazado las rutas según su propia experiencia y habilidad, por tanto sin garantía de que la ruta definida sea la que conduce a la menor compactación del suelo y al menor consumo en combustible, situación que se hace más grave cuanto más extensos son los cultivos. En general, el criterio que siempre se ha seguido es el de reducir la distancia recorrida por su estrecha relación con la compactación, el tiempo y el combustible consumido, que es a su vez la razón por la que la mayoría de los trabajos científicos publicados hasta ahora abordan el problema de planificación de ruta con un criterio de mínima distancia. Así (Stoll, Stafford, & Werner, 2003) proponen descomponer el campo en sub-campos más pequeños y con formas más simples, planificando cada una de las partes para a continuación conectar las soluciones de tal modo que la distancia sea mínima. Una de las ventajas de este planteamiento es que, por construcción, se evitan los solapamientos y las zonas sin tratar, ya que cada parte se visita exactamente una única vez. Esto se asemeja a la estrategia ya estudiada en (Conesa-Muñoz & Ribeiro, 2012) donde se descomponía el espacio en triangulaciones. No obstante, esta forma de segmentación se adapta mejor a planificaciones en espacios sin estructurar, en los que es posible moverse en cualquier dirección –siempre que no haya un obstáculo–, mientras que en el contexto agrícola, al estar los campos compuestos por líneas de cultivo, como veremos más adelante, el espacio está estructurado con un patrón claramente definido y repetitivo, lo que permite que existan otras estrategias más adecuadas para encontrar la mejor trayectoria.

La mayoría de los trabajos emplean segmentaciones espaciales de un modo u otro, aunque utilizando descomposiciones geométricas más adecuadas que las anteriores. Por ejemplo, en (Sorensen, Bak, & Jorgensen, 2004) se divide el campo en sub-campos y cada uno de ellos se representa como un grafo donde los vértices se sitúan en los contornos y las aristas lo atraviesan de tal modo que si se recorren todas se cubre toda la superficie. El trabajo acaba reformulando el problema como el problema clásico del cartero chino (CPP) (Mohammaditabar, 2013), que consiste en encontrar el camino que pasa por todas las aristas, y aborda su solución con un método voraz.

En (Taïx, Souères, Frayssinet, & Cordesses, 2006) se divide el campo en polígonos convexos, se localiza las áreas de giro y se usa la dirección de siembra como dirección para recorrer el cultivo dentro de cada polígono. Una vez dividido el campo en polígonos, cada uno de ellos se representa mediante un grafo, pero en esta ocasión son los vértices los que se disponen de tal modo que deben visitarse todos para cubrir el campo por completo. El problema se reformula entonces para encontrar el mejor camino *hamiltoniano* (camino que pasa por todos los vértices de un grafo) mediante un algoritmo voraz que selecciona en cada paso el siguiente punto más prometedor según una heurística basada en la distancia.

En (Oksanen & Visala, 2009) también se descompone el campo en partes y se proponen dos algoritmos basados en heurísticas. El primero divide el campo en trapecoides y busca el mejor orden para cubrirlos todos. En este caso, encontrar el orden

óptimo no es muy costoso computacionalmente porque suele ser suficiente con dividir el campo en pocas partes. Después, para cada trapezoide, se buscan las rutas internas usando el método heurístico *boustrophedon* (Choset & Pignon, 1998) que no garantiza encontrar el óptimo. El segundo algoritmo se centra más en campos con límites curvos en los que no es fácil aplicar la descomposición trapezoidal del método anterior. En este caso la estrategia que se propone consiste en seguir el contorno del campo, incluso cuando no es recto, habiéndolo dividido previamente en franjas y recorrerlo enlazando cada franja con su adyacente. Los propios autores indican que para obtener la solución óptima la siguiente franja no tiene por qué ser la más cercana, sino que teóricamente habría que evaluar todas las alternativas posibles, seleccionar una, y repetir el proceso recursivamente con las restantes hasta no quedar ninguna. Como este proceso es inabordable por el gran número de soluciones posibles, lo que los autores proponen es determinar en cada paso la franja mejor entre un conjunto más limitado, esto es, una búsqueda local en la que solo se evalúan las franjas más cercanas. De este modo se acelera la búsqueda aunque sin garantizar la obtención del óptimo.

En todos estos trabajos los métodos empleados usan estrategias voraces que seleccionan en cada paso la opción que parece mejor según una heurística, pero esta forma de proceder no garantiza la obtención del máximo global (óptimo) cuando existen máximos locales. Si se cae en un máximo local y siempre se selecciona la mejor opción, es imposible pasar por los pasos intermedios peores necesarios para escapar del máximo local y tener una oportunidad para llegar al global.

En cualquier caso, todos estos trabajos muestran que la planificación en campos agrícolas puede expresarse como un problema de optimización combinatoria en el que debe encontrarse el orden óptimo para conectar todas las partes en las que se ha dividido el campo (ya sean sub-campos, polígonos, trapezoides, franjas...). De hecho, como la distancia es el criterio que se usa para determinar el óptimo, el problema de planificación de rutas es análogo al problema del viajante o TSP (por *Traveling Salesman Problem* en inglés). EL TSP es un problema clásico dentro de la optimización combinatoria (Gutin & Punnen, 2002). Consiste en encontrar la ruta más corta que debe seguir un viajante para visitar una única vez n ciudades retornando a la ciudad inicial una vez visitadas todas. En el presente contexto agrícola, el problema es análogo salvo que, en vez de visitar ciudades, se visitan y trabajan las diferentes partes en las que se divide el campo y no se requiere volver al punto de partida.

Como el objetivo final es obtener la ruta más corta y en el contexto agrícola se trabaja con vehículos, es absolutamente necesario tener en cuenta los radios de giro de éstos para trabajar con trayectorias reales y obtener resultados realistas. En (Bochtis & Vougioukas, 2008) se estudia cómo afecta el radio de giro de un vehículo a las trayectorias en un campo agrícola. Al igual que en los trabajos anteriores, se descompone el campo en franjas paralelas, llamadas calles, orientadas según la dirección de cultivo y de ancho igual al ancho de trabajo de la herramienta que se utiliza para realizar la tarea agrícola. Posteriormente, una vez más, se busca el mejor orden para recorrerlas todas. Para encontrar este orden, se explica que todos los vehículos tienen un mínimo radio de giro determinado por las limitaciones mecánicas de la maquinaria. A partir de este mínimo radio de giro se muestran los tipos de maniobras que pueden darse para pasar de una calle a otra en un campo de forma rectangular. Según el ancho de las calles, su separación, la longitud del radio y la posible existencia de un obstáculo entre medias, pueden darse las tres maniobras que se muestran en la Figura 2.1.

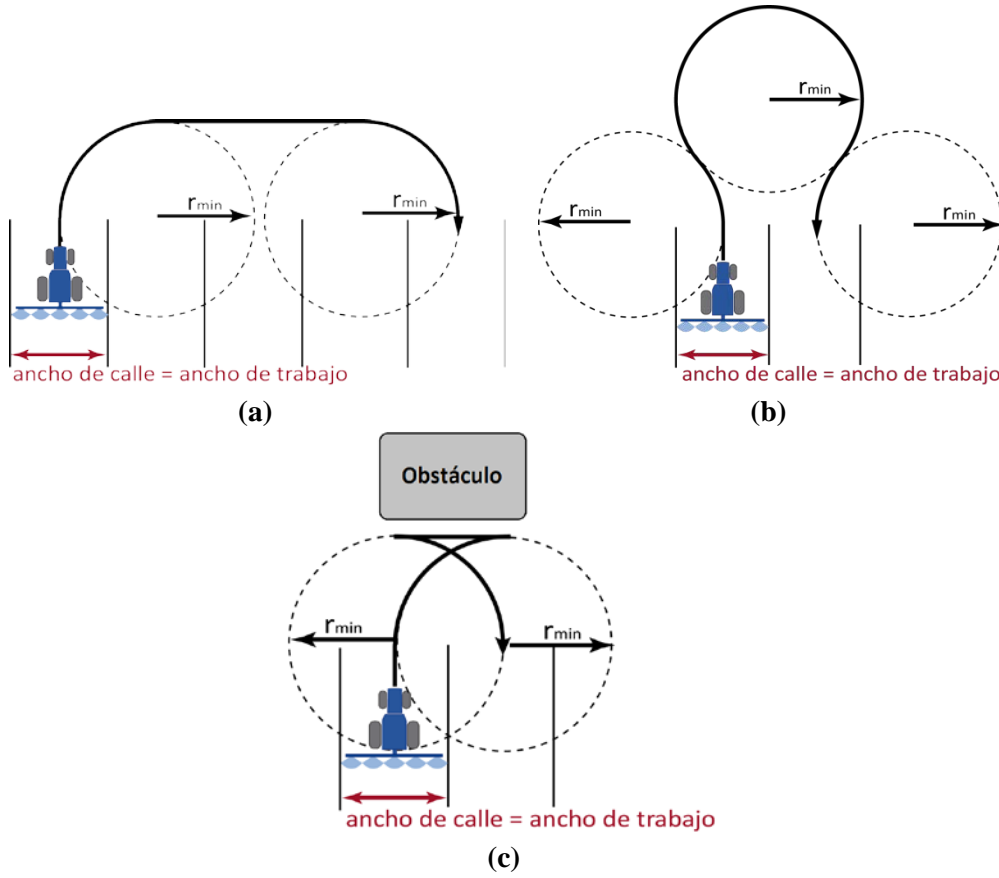


Figura 2.1. Tipos de maniobra para un vehículo de radio r_{min} en un campo regular. (a) Π -giro, (b) Ω -giro y (c) T-giro.

La maniobra Π (Figura 1a) se emplea cuando hay suficiente distancia entre calles para poder transitar en dos giros de 90° , en caso contrario se utiliza la maniobra Ω (Figura 1b). El giro T (Figura 1c) involucra desplazamiento marcha atrás por lo que es un giro complejo que solo se usa en aquellos casos en los que hay algún obstáculo y no se pueda ejecutar ninguna de las maniobras anteriores por falta de espacio. En general, los campos se cultivan dejando en las cabeceras el espacio libre suficiente para poder realizar los giros Π y Ω , por lo que el giro T no es una maniobra habitual y por ende no se considera en el resto de la memoria.

Las ecuaciones (2.1) y (2.2) calculan la distancia para las maniobras Π y Ω entre las calles i y j . Mientras que la ecuación (2.3) indica, a partir del radio de giro r_{min} de un vehículo y el ancho de calle a_c , cuando se debe aplicar un giro u otro.

$$\Pi(i, j) = |i - j|a_c + (\pi - 2)r_{min} \quad (2.1)$$

$$\Omega(i, j) = r_{min} \left(3\pi - 4 \sin^{-1} \left(\frac{2r_{min} + |i - j|a_c}{4r_{min}} \right) \right) \quad (2.2)$$

$$giro(i, j) = \begin{cases} \Pi(i, j), & \text{si } |i - j|a_c \geq 2r_{min} \\ \Omega(i, j), & \text{e. o. c. } |i - j|a_c < 2r_{min} \end{cases} \quad (2.3)$$

Cuando el radio de giro del vehículo no es pequeño, recorrer un campo continuando siempre por la franja adyacente no es siempre la opción más óptima aunque sea la más sencilla para el operario, ya que puede contabilizar mejor lo que ha hecho y lo que le queda por hacer. De la Figura 2.1b se desprende que con este tipo de giro las distancias recorridas son mayores y en consecuencia se tardará más en cubrir el campo y se consumirá más combustible. En su propuesta Bochtis & Vougioukas tienen en cuenta lo anterior y calculan las distancias entre todas las calles del campo usando las ecuaciones (2.1), (2.2) y (2.3) y emplean un algoritmo heurístico de ahorros (Clarke & Wright, 1964) para obtener una solución aproximada que, como en otros casos, no se garantiza que sea la óptima.

Puesto que el TSP es un problema NP-completo (Gutin & Punnen, 2002), no se conoce ningún algoritmo que sea capaz de encontrar la mejor solución en tiempo polinómico. El tamaño del espacio de búsqueda de este problema es enorme y crece considerablemente con cada nueva ciudad. Si hay n ciudades, el número de soluciones posibles es $(n - 1)!$ En efecto, partiendo de una ciudad, en el primer paso tenemos que seleccionar una de las $n - 1$ ciudades restantes, en el segundo entre las $n - 2$ restantes, etc. Por ejemplo, para solo 20 ciudades hay 121.645.100.408.832.000 soluciones posibles, un número tan grande que serían necesarios 4 millones de años para explorar todo el espacio de soluciones si se tardara 1 ms en inspeccionar cada solución. Esta es la razón de que hasta el momento muchos trabajos han abordado el problema de planificación de rutas en entornos agrícola con métodos voraces guiados por heurísticas que, aunque no garantizan el óptimo, ofrecen soluciones relativamente cercanas en muchos casos y en un tiempo razonable. Más recientemente algunos autores han propuesto métodos de optimización combinatoria, más lentos y sofisticados que han permitido alcanzar soluciones mejores. Por ejemplo, en (Ryerson & Zhang, 2007) se usa un Algoritmo Genético para saber cómo cubrir un campo con obstáculos cuando no es obligatorio mantener una dirección de trabajo, es decir, cuando el vehículo puede moverse libremente en cualquier dirección. Descompone el campo en celdas, como si de una matriz se tratase, y etiqueta como obstáculos las ocupadas por estos. Posteriormente el Algoritmo Genético se encarga de encontrar el mejor orden para recorrer todas las celdas libres. Como comentan los autores en las conclusiones, se trata de una primera aproximación que no considera parámetros importantes como el radio de giro o la velocidad del vehículo. Es más, la estrategia no es aplicable a cultivos ya establecidos donde la maquinaria tiene que moverse siguiendo la dirección del cultivo para no dañarlo y, por tanto, la propuesta tiene fundamentalmente interés previamente a la siembra para determinar la mejor forma para establecer el cultivo. Es importante resaltar que, una vez establecido el cultivo, es la dirección de siembra la que utilizan los agricultores para recorrer el campo, ya que otra dirección no garantiza que no se dañen las plantas de cultivo. Es más, el moverse en una dirección diferente provoca la aparición de más vibración en la maquinaria como efecto del relieve de los surcos cuando estos se atraviesan por las ruedas del vehículo.

En (Hameed, Bochtis, & Sorensen, 2011), también se aplica un Algoritmo Genético en dos etapas: en la primera, previa a la siembra, se determina la dirección en la que debe establecerse el cultivo y, en la segunda, se divide el campo en calles que siguen la dirección establecida para determinar el mejor orden para recorrerlas todas. En ambos casos el objetivo es minimizar la distancia que recorre el vehículo en las zonas de

transición (cabeceras). Las soluciones obtenidas no son óptimas pero se muestran más eficientes que los patrones convencionales de trabajo en campo.

La planificación de rutas se vuelve más compleja cuando la tarea se lleva a cabo con más de un vehículo. Aunque, como ya se comentó anteriormente en la sección 1, el uso de flotas de vehículos tiene muchas ventajas, también conlleva la utilización de modelos más complejos (Sørensen & Bochtis, 2010) que tienen que considerar aspectos como la existencia de más elementos activos en movimiento, los riesgos de colisión, la posibilidad de duplicidades en la realización de la tarea, etc.

Cuando se utiliza una flota, el problema es análogo a la extensión del TSP para varios viajantes conocida como mTSP o *multi TSP* un problema, al igual que el TSP, NP-completo. Así se formula el problema de la generación de rutas para una flota de cosechadoras en (Bochtis D. , Vougioukas, Tsatsarelis, & Ampatzidis, 2007), dónde el campo se divide en calles y se ataca el problema utilizando un algoritmo de ahorros (Clarke & Wright, 1964).

Cuando los viajantes del mTSP son vehículos, como sucede en el problema agrícola, es más frecuente oír hablar del problema de rutas de vehículos (en inglés *Vehicle Routing Problem*, VRP). En este problema se usa una flota de camiones para transportar bienes desde un depósito a una serie de clientes a los que hay que suministrar además una cantidad determinada (Dantzig & Ramser, 1959). Todos los clientes deben ser visitados exactamente una vez y deben recibir el número de bienes que demandan, que no tiene que ser el mismo para todos. Además, los vehículos deben empezar y acabar su ruta en el depósito. El objetivo del problema consiste en encontrar las rutas que permitan que la flota realice el reparto recorriendo en conjunto la mínima distancia. En otras palabras, el VRP puede entenderse como una extensión del mTSP en la que, además de visitar a los clientes, los viajantes tienen que entregar un número determinado de bienes a cada cliente. En el contexto agrícola, los clientes son las diferentes partes en las que se divide el campo y los bienes asignados a cada cliente son el insumo requerido por la tarea en cada parte. Al igual que sucedía en el caso del TSP en su versión agrícola, no es necesario que los vehículos partan y vuelvan al mismo punto, es decir, al depósito.

En (Bochtis & Sørensen, 2009) y (Bochtis & Sørensen, 2010) se formula el problema de planificación agrícola en estos términos y se hace una clasificación teórica de con que variante del VRP encaja cada trabajo agrícola según las condiciones en las que se ejecuta. Esto es, si se utiliza un solo vehículo se habla de TSP, si cada parte debe ser trabajada dentro de un intervalo de tiempo concreto se habla de TSPTW (TSP *with time windows*) o incluso de VRPTW (VRP *with time windows*) en caso de disponer de una flota de vehículos, o si se desconoce el volumen de insumo requerido por la tarea a realizar se habla de DVRP (*dynamic VRP*), y así con muchos otros casos.

De entre todas las variantes que se nombran en ambos artículos, las que son la base de la propuesta realizada en el presente trabajo de investigación por ser las más interesantes son: 1) la que considera que los vehículos tienen una capacidad limitada de almacenamiento/transporte (CVRP, en inglés *capacited VRP*) y 2) la que permite que los vehículos sean heterogéneos, es decir, que dentro de la flota las distintas unidades puedan tener distintas características como distinta capacidad de carga, velocidad de trabajo, consumo de combustible, etc. A esta última variante, se la llama MFVRP (en inglés *mixed fleet VRP*). Estas variantes son las más interesantes porque son las que más cerca se encuentran de la planificación en la práctica real.

Respecto a la heterogeneidad de la flota, hasta la fecha ninguna aproximación en el contexto agrícola ha considerado vehículos con diferentes características, aún siendo

bastante habitual que un agricultor posea unidades con diferentes consumos de combustible, capacidades de carga, capacidades del tanque de combustible, etc. Así la generalización utilizada hasta el momento obliga a los agricultores a trabajar usando solo vehículos similares o, cuanto menos, a usarlos en las condiciones de trabajo de los peores vehículos de la flota para así igualar sus características y satisfacer las condiciones del problema. Por ejemplo, si un agricultor dispusiera de dos vehículos, el primero con capacidad de almacenamiento de 100 unidades y velocidad de trabajo de 6 km/h, y el segundo de 200 unidades y 3 km/h respectivamente, para cumplir las condiciones del CVRP habría que forzar a la flota a trabajar con una capacidad de 100 unidades y una velocidad de 3 km/h, situación claramente ineficiente.

Aunque en el contexto agrícola no hay trabajos que aborden la heterogeneidad en la flota, en otros ámbitos sí que existen trabajos que se enfrentan a problemas con flotas de vehículos heterogéneas (Baldacci, Battarra, & Vigo, 2008)(Lee, Kim, Kang, & Kim, 2008) (Salhi, Imran, & Wassan, 2014).

Respecto a la limitación de capacidad de carga o del tanque, esta restricción es muy importante porque fuerza a los vehículos a desplazarse hasta el depósito, que es el punto habitual de carga y descarga (a veces también el de salida y llegada), para reabastecerse cuando agotan algún insumo (combustible, agroquímico, etc.) o cuando están cosechando y se quedan sin espacio de almacenamiento y necesitan descargar el grano. Es habitual que los vehículos consuman una determinada cantidad de insumo a medida que visitan las zonas y, por tanto, la longitud de las rutas está limitada por la cantidad de recursos que los vehículos pueden transportar. Por ejemplo, si la tarea consiste en un tratamiento por pulverización de herbicida, las zonas demandan herbicida y los vehículos lo transportan y lo suministran. Si la tarea consiste en cosechar, la demanda puede entenderse como la cantidad de grano que los vehículos deben llevarse. En (Oksanen & Visala, 2009) se estima que un vehículo suele necesitar repostar herbicida cada 4 ha, o una cosechadora depositar el grano cada 0,5 ha, por lo que estas operaciones no son casos excepcionales sino que pueden suceder con relativa frecuencia durante la ejecución de una tarea en un cultivo real.

Ya que no solo hay que contar el desplazamiento de ida del campo al depósito sino también el de vuelta del depósito al campo para continuar con la tarea, la distancia puede verse incrementada considerablemente, aumentando el tiempo y el combustible empleados. A pesar de la importancia de esta cuestión, pocos trabajos tienen en cuenta esta limitación, y cuando lo hacen solo consideran un vehículo (Oksanen & Visala, 2009), (Bochtis, Sørensen, & Vougioukas, 2010). Resulta interesante la propuesta de (Bochtis, Sørensen, & Vougioukas, 2010) en la que se utiliza una unidad móvil secundaria que se desplaza en las cabeceras para el abastecimiento (combustible y agroquímico) o la recogida (cosecha), una buena estrategia que disminuye el espacio de ida y vuelta recorrido fuera del cultivo pero que no afecta al recorrido realizado dentro del cultivo por lo que a pesar de la unidad secundaria sigue siendo esencial planificar para minimizar el número de cargas/descargas necesarias.

Otro de los aspectos mejorables en la planificación es que la mayoría de las aproximaciones fijan la distancia recorrida como único criterio a tener en cuenta, aunque otros criterios como el tiempo para terminar la tarea o los insumos (combustible, agroquímico, etc.) utilizados, también deberían considerarse por el impacto que tienen sobre el coste económico de la tarea. El tiempo influye en el coste ya que involucra el salario de los trabajadores (Pedersen, Fountas, Have, & Blackmore, 2006), mientras que el insumo penaliza por su coste y por el posible daño al medioambiente (Berny, 2007) de

algunos de los productos utilizados, lo que se acaba traduciendo en más gastos de reparación. Aun así, hasta la fecha, existen muy pocos trabajos que hayan considerado otros criterios que no sean la distancia recorrida. A partir de la distancia y de las velocidades de trabajo es fácil adaptar las aproximaciones ya citadas para que usen el tiempo como criterio de optimización, es más, algunos pocos trabajos ya utilizan el tiempo directamente, expresándolo en función de la distancia y la velocidad (Jensen, Bochtis, Sørensen, Blas, & Lykkegaard, 2012), (Bochtis, Sørensen, Green, Moshou, & Olesen, 2010), (Sørensen & Nielsen, 2005). No obstante, no hacen distinción entre las velocidades de trabajo o velocidades en el interior del cultivo, limitadas por las características de la tarea y el apero, y las velocidades en las zonas de transición (cabeceras) habitualmente sin límite, por lo que los resultados no son del todo realistas.

Respecto a los insumos, en (Gonzalez-de-Soto, Emmi, Garcia, & Gonzalez-de-Santos, 2015) y (Hameed, 2014) se proponen aproximaciones para calcular el consumo de combustible de un vehículo en movimiento en función de un mapa de elevación del terreno. Los autores utilizan la estimación para determinar la mejor dirección de movimiento en términos de consumo de combustible. En (Gonzalez-de-Soto, Emmi, Garcia, & Gonzalez-de-Santos, 2015) se supone que el vehículo no tiene restricciones de movimiento dentro del campo una vez establecido el cultivo, lo que es una situación poco realista como ya se ha comentado anteriormente. Ahondando en la idea de planificar optimizando otros criterios además de la distancia recorrida, un aspecto importante es el de cómo combinar los distintos criterios y este punto es especialmente complejo de abordar cuando se desea encontrar una solución mínima que considera dos criterios diferentes entre los que existe una relación opuesta, es decir, cuando uno aumenta el otro disminuye y viceversa. Un ejemplo puede ser el de un agricultor interesado en minimizar el tiempo de ejecución de una tarea en el campo y que quiere alquilar la maquinaria necesaria, incluidos los vehículos. Cuantos más vehículos utilice más rápidamente realizará la tarea pero también el coste del alquiler será más alto. La mejor solución tiene que encontrar, por tanto, un compromiso entre el tiempo necesario para realizar la tarea y el coste de la maquinaria. En otros términos, lo que se está planteando es una optimización multi-objetivo que hasta ahora no se ha estudiado en planificación agrícola, pero que sí se ha utilizado y es bien conocida en otros muchos contextos (Deb, Multi-objective Optimization, 2014).

Por último, otro factor que puede afectar a la planificación es la variabilidad del cultivo. En los últimos años han proliferado mucho los mapas de cultivos (Oliver M. A., 2010). Estos mapas son muy útiles porque muestran las variabilidades existentes en el cultivo (qué partes son más productivas, cuáles requieren más agua, cuál es la distribución de los rodales de malas hierbas, etc.), información que puede usarse para llevar a cabo las tareas agrícolas más eficientemente. Por ejemplo, como se vio en la introducción, por término medio es posible reducir en más de un 70% el herbicida utilizado en tratamientos de control de malas hierbas, simplemente acotando el tratamiento a las zonas infestadas (Slaughter, Giles, & Downey, 2008).

A tenor de todo lo expuesto, resulta evidente la necesidad de desarrollar un planificador más general, que pueda planificar las rutas a seguir en multitud de situaciones agrícolas, para uno o varios vehículos a la vez con características heterogéneas, con limitaciones físicas reales (radio de giro, capacidad de almacenamiento), considerando la posibilidad de reabastecimiento, teniendo en cuenta la variabilidad del cultivo (principalmente la distribución de malas hierbas), que trabaje con cualquier forma de campo e incluso con campos compuestos por varias parcelas, y, por

último, que sea capaz de optimizar diferentes criterios, bien combinándolos adecuadamente en un único objetivo o mediante optimizaciones multi-objetivo. A lo largo de la tesis se explicará el trabajo llevado a cabo para implementar dicho planificador.

2.3. Supervisión de la ejecución de la tarea

Una vez realizada la planificación, el siguiente paso consiste en asignar a cada robot de la flota un plan de trabajo con la ruta a recorrer además de las operaciones específicas a realizar en cada zona, para que entre todos realicen la tarea autónomamente. La labor autónoma de los robots en el campo debe ser supervisada en todo momento porque cualquier malfuncionamiento o pequeña desviación del plan previsto puede tener graves consecuencias, máxime si tenemos en cuenta que, aún en el caso de robots agrícolas pequeños/medianos, estamos hablando de vehículos con cargas que rondan los 500 kg y, aunque se mueven a velocidades bajas, entre 3 y 8 km/h, son potencialmente peligrosos. A esto hay que sumar que los entornos agrícolas son complejos por ser muy variables en el tiempo lo que hace que solo puedan ser parcialmente conocidos. Así, poseen una incertidumbre inherente a la imposibilidad de disponer con antelación información precisa sobre aspectos importantes como el estado del terreno, la temperatura, la iluminación, la humedad, etc. A todo esto hay que añadir la posible existencia de operarios trabajando en las inmediaciones o la posibilidad de que aparezcan animales repentinamente. Por tanto es previsible que la ejecución varíe de lo previamente planificado, por lo que resulta complejo automatizar completamente la tarea agrícola manteniendo las condiciones de seguridad necesarias para las personas, la maquinaria y el propio cultivo (Edan, 2009). Si a esto se añade que uno de los objetivos de la presente tesis es que la tarea se lleve a cabo mediante varios vehículos trabajando simultáneamente y con un único operario supervisándolos, se hace indispensable disponer de un sistema de supervisión de ayuda al operario que sea capaz de interpretar el entorno de los vehículos de la flota en tiempo real, que compruebe que la tarea se está ejecutando como corresponde a pesar de la incertidumbre y que, en caso de no ser así, detecte los problemas cuanto antes e identifique las causas con precisión. En el capítulo anterior se mencionaron sistemas de control de vehículos y flotas que en mayor o menor grado supervisan y se enfrentan a los efectos de la incertidumbre como por ejemplo los sistemas que rectifican la posición del vehículo (Sistemas agrícolas avanzados CIH, 2015)(John Deere Guidance Systems, 2015)(Trimble GNSS Systems, 2015) o la inclinación/altura del apero(Pulverizador remolcado serie M700/M700i John Deere) o los sistemas programados que pulverizan el herbicida a un flujo constante. Sin embargo se trata de sistemas relativamente sencillos que no abordan el problema de forma global, esto es, monitorizando varios aspectos a la vez y para una flota de vehículos que cubren el cultivo realizando la tarea encomendada de una forma totalmente autónoma. La gran cantidad de información que hay que analizar en este escenario, sobre la que además hay que reaccionar en tiempo real, hace que el problema se vuelva muy complejo técnicamente. No obstante, al margen del ámbito agrícola existen sistemas y planteamientos que resuelven la cuestión atacando de forma general la fuente de la desviación entre la planificación y la ejecución, en otras palabras, la incertidumbre.

Dentro del campo de los robots autónomos, (Saffiotti, 1999) propone tres estrategias diferentes, para manejar la incertidumbre y sus efectos, que pueden aplicarse

perfectamente en el desarrollo de maquinaria agrícola autónoma, a saber, (1) eliminar la incertidumbre, (2) tolerarla y (3) razonar sobre ella.

La primera estrategia se basa en mejorar los sistemas que introducen la incertidumbre, por ejemplo usando mejor instrumentación, esto es, más precisa y que ofrezca más información. En esta categoría, dentro de las plataformas agrícolas, entrarían las ya mencionadas soluciones con receptores GPS de alta precisión o láseres. El problema de esta estrategia es que encarece considerablemente el coste de las plataformas. Además, no solo los dispositivos generan incertidumbre, también el entorno puede hacerlo, por lo que conviene estructurarlo en la medida de lo posible, por ejemplo incluyendo puntos de referencia o pistas fijas en el terreno con las que guiar y tener mayor certeza de los movimientos de los robots (Hague, Marchant, & Tillett, 2000). Sin embargo, estas medidas reducen la flexibilidad y robustez de las soluciones, porque las hacen muy dependientes del estado de las referencias o las pistas que pueden presentar problemas tales como estar tapadas por un obstáculo, estar rotas, etc.

En cualquier caso, no siempre se pueden eliminar todas las fuentes de incertidumbre mediante alguno de los métodos anteriores. En el campo siempre hay que contar con la posible aparición de obstáculos móviles como son los operarios, situación poco frecuente si se controla la entrada a la zona de trabajo, y los animales, situación más frecuente en algún tipo de cultivo. Es por eso que la segunda estrategia propuesta por (Saffiotti, 1999), tolerar la incertidumbre, se traduce en que una plataforma autónoma debe estar preparada para manejar las situaciones con incertidumbre que surjan durante su funcionamiento. En otras palabras, es necesario que la plataforma incorpore algún tipo de sistema supervisor capaz de monitorizar su funcionamiento y así controlar los efectos de la incertidumbre. Estos sistemas suelen estar basados en modelos muy ambiguos, para que casi cualquier situación tenga cabida en ellos y pase lo que pase nunca dejen de operar dentro del marco de trabajo esperado.

La tercera estrategia, razonar sobre la incertidumbre, es la de más alto nivel y por tanto la más compleja. Consiste en analizar de algún modo lo que está sucediendo, contrastarlo con algún modelo más o menos preciso o experiencia previa que pueda arrojar un poco de luz al asunto, razonar las diferencias e inferir alguna solución con la que actuar convenientemente. Sin embargo, los modelos precisos no son siempre fáciles de obtener, a veces son incluso imposibles, especialmente cuando los comportamientos de los sistemas son complejos por haber muchos casos posibles. En estas situaciones es muy difícil abarcarlo todo, e incluso así no todo puede resolverse razonando. Ningún razonamiento puede prever con total certeza lo que va a suceder cuando no se dispone de información. Un robot puede suponer lo que está detrás de una puerta, pero no puede saber con total garantía lo que hay hasta que se abra. De ahí que, aunque sea recomendable razonar sobre la incertidumbre, siempre se deba estar preparado para tolerarla.

De todo esto se deduce que la mejor estrategia consiste en combinar los tres planteamientos anteriores en mayor o menor medida (Saffiotti, 1999). En definitiva, disponer de un sistema que monitorice la ejecución de la tarea trabajando sobre un entorno parcialmente predecible, con una instrumentación precisa que introduzca poca incertidumbre, pero que, cuando lo haga, se pueda tolerar y razonar sobre ella para contrarrestarla en la medida de lo posible.

A la mayoría de los sistemas de supervisión de este tipo se les exigen dos funcionalidades principales, indispensables también en el contexto agrícola: detección y diagnóstico de fallos (Gertler, 1998). La primera funcionalidad detecta si algo va mal y la

segunda estima su gravedad. En (Chiang, Braatz, & Russell, 2001) se distinguen tres tipos de esquemas para implementar sistemas basados en estos dos aspectos: analíticos, guiados por datos y basados en conocimiento.

Los analíticos usan modelos matemáticos del proceso real y tienen dos etapas. En la primera, las entradas y las salidas medibles del sistema se comparan con un modelo que describe cuál debería ser la relación entrada-salida en términos matemáticos exactos. Cualquier inconsistencia que se detecte con respecto a la relación indicará que hay un fallo en el sistema. En la segunda etapa, se hace el diagnóstico del fallo examinando las inconsistencias detectadas.

Este tipo de esquemas son preferibles cuando el sistema a monitorizar es bien conocido y la incertidumbre es escasa. Sin embargo, no siempre hay modelos matemáticos disponibles o a veces es muy difícil o imposible obtenerlos. Ésta es la principal debilidad de esta opción que, aun así, se ha usado asiduamente para la supervisión de trayectorias en plataformas móviles autónomas (Williams & Nayak, 1996)(Simmons, Fernandez, Goodwin, Koenig, & O'Sullivan, 2000)(McCarthy & Pollack, 2000).

Por el contrario, los esquemas guiados por datos derivan directamente la detección y el diagnóstico de los datos de entrada, generalmente a partir de un proceso de toma de decisiones a menudo basado en métodos estadísticos. Estos sistemas están pensados para monitorizar sistemas industriales de gran escala, como plantas industriales enteras o robots autónomos, es decir, plataformas con mucha instrumentación que, por tanto, son capaces de producir una enorme cantidad de datos. Asimismo, a partir de técnicas de data mining es posible inferir información de control con la que gobernar mejor el propio proceso, tal como se hace en algunos robots autónomos (Dearden, Willeke, Simmons, Verma, Hutter, & Thrun, 2004)(Pettersson, Karlsson, & Saffiotti, 2004). No obstante el rendimiento de estos sistemas es altamente dependiente de la calidad y el volumen de los datos de entrada.

Por último, el objetivo de los esquemas basados en conocimiento es simular la conducta de un experto resolviendo problemas y tareas en un contexto de supervisión. El punto fuerte de esta opción es que permite combinar las anteriores construyendo sistemas híbridos que se adaptan mejor a los entornos reales. Vehículos no tripulados que usan esta metodología se encuentran en los trabajos de (Pettersson, Karlsson, & Saffiotti, 2005) y (Ranganathan, Patel, & Sathyamurthy, 2001).

Todos estos sistemas de supervisión, capaces de detectar y diagnosticar fallos, pueden complementarse con una tercera funcionalidad para cerrar el lazo de control, en otras palabras, con la reparación (Chiang, Braatz, & Russell, 2001). Este paso es bastante complejo pero muy necesario en plataformas totalmente autónomas en las que la intervención humana no es posible, como sucede con las plataformas de exploración planetaria (Washington, Golden, Bresina, Smith, Anderson, & Smith, 1999), o no es lo suficientemente rápida para evitar el problema, como pasa en los sistemas de conducción autónoma (Brandstotter, Hofbaur, Steinbauer, & Wotawa, 2007).

En el ámbito agrícola, un sistema de supervisión debería monitorizar las trayectorias, las velocidades, el estado de los aperos (si están actuando correctamente), además de las dificultades añadidas que pueden surgir por el control de toda una flota y no de un único vehículo. Por ejemplo, los riesgos de colisión entre los diferentes miembros de la flota, o incluso los bloqueos que puedan aparecer porque uno de los vehículos cierre el paso a otro. En la actualidad no existe ningún sistema de supervisión de tareas agrícolas de tan alto nivel que reúna las características mencionadas. Existen

algunos que, como ya se ha mencionado, monitorizan parcialmente la trayectoria o el flujo de herbicida, pero no existen sistemas capaces de supervisar en tiempo real todos los aspectos a la vez y para varios vehículos trabajando simultáneamente de forma totalmente autónoma.

Un sistema de estas características podría garantizar que un único operario fuera capaz de controlar toda la flota. Le ayudaría detectando las anomalías que surgieran durante la ejecución de las tareas, tales como posiciones incorrectas, velocidades de trabajo inadecuadas, aperos no activados a tiempo, riesgos de colisión, etc. El sistema notificaría convenientemente al operario el problema para que éste buscara la mejor forma de resolverlo, incluso en algunos casos concretos podría resolver el problema sin intervención del operario.

Un sistema Supervisor de este tipo supondría una valiosa contribución hacia la automatización de la agricultura mediante flotas de robots, de ahí que su desarrollo haya sido uno de los objetivos principales de la presente tesis.

2.4. Separación de la tarea agrícola en inspección y tratamiento

En muchos problemas medioambientales es necesaria una primera etapa de inspección sobre el terreno para poder simplificar o ayudar a la siguiente etapa de intervención. En multitud de ocasiones la zona afectada es grande y de acceso limitado por lo abrupto del terreno y por los propios obstáculos naturales, por lo que la inspección desde tierra no es viable y los vehículos aéreos no tripulados aparecen como una solución idónea para inspeccionar estas zonas. La resolución de estos problemas puede descomponerse entonces en dos etapas: 1) una inspección aérea mediante drones y 2) una intervención terrestre mediante plataformas móviles grandes que en algunos casos tiene que transportar grandes equipos. La inspección aérea puede suministrar una estimación rápida y fiable de las zonas afectadas, que puede utilizarse en la intervención terrestre para llevar a cabo el trabajo de modo más eficiente. Una estrategia bien integrada de robots aéreos y terrestres conduciría a un empleo más eficiente de los robots autónomos utilizados actualmente en catástrofes medioambientales, por ejemplo, en vertidos de petróleo en el mar (Gaudin, 2010), incendios forestales (Casbeer, Beard, McLain, Li, & Mehra, 2005) o terremotos (Nathan, y otros, 2012).

La agricultura también puede beneficiarse de esta práctica, ya que, para planificar y ejecutar las tareas agrícolas eficientemente, no basta con prestar atención solo a la geometría del campo y a las características de la maquinaria, sino que también es útil conocer el estado del propio cultivo ya que las tareas agrícolas a realizar pueden ser más eficientes si se considera la variabilidad del mismo; por ejemplo, como ya se ha apuntado anteriormente, en un hipotético tratamiento variable de control de plagas, se trata de saber cuál es el grado de infestación de cada zona y administrar el fitosanitario en la dosis recomendada.

Precisamente en la agricultura, gracias a la automatización y a los numerosos avances en los sistemas de percepción y en los propios sensores, ahora es más fácil extraer este conocimiento (Mulla, 2013), generalmente en forma de mapas de variabilidad. Las misiones de inspección se han simplificado mucho y ahora una única plataforma puede realizar los muestreos de campo que antes se hacían entre varias

personas y de forma tediosa (Fishel, y otros, 2001). Esto ha propiciado que las tareas agrícolas se desacoplen a alto nivel en las ya mencionadas fases de inspección e intervención o tratamiento

En el caso de los escenarios agrícolas, no es tan necesario que la inspección se lleve a cabo mediante unidades aéreas, ya que los terrenos del cultivo no son muy irregulares y están preparados para contener los mínimos obstáculos posibles. Además, aunque la inspección aérea presente grandes ventajas como una gran cobertura en poco tiempo y sin impacto en el cultivo, la inspección terrestre permite una mayor resolución con la posibilidad, además, de utilizar equipos más sofisticados formados por más de una cámara. Lo que sí está claro es que si se separan las actuaciones en una etapa de inspección y otra de tratamiento, la inspección debe llevarse a cabo mediante pequeñas plataformas, baratas, ligeras y con poco impacto sobre el cultivo; dejando la etapa de tratamiento para plataformas móviles más grandes, con la capacidad suficiente para transportar las herramientas de trabajo y los insumos en caso de ser necesario. De ahí que en este caso las plataformas sean más pesadas.

Son numerosos los trabajos que se apoyan en esta idea y abordan alguna de las dos etapas por separado. Por ejemplo, para la inspección se han propuesto plataformas móviles terrestres (Bak & Jakobsen, 2004) y aéreas (Barrientos, y otros, 2011) para construir mapas de malas hierbas (Peña-Barragán, Torres-Sánchez, de Castro, Kelly, & López-Granados, 2013), de cantidad de cosecha (Stafford, Ambler, Lark, & Catt, 1996) o de la fertilidad del suelo (Cahn, Hummel, & Brouer, 1994) entre otros muchos.

Del mismo modo, la agricultura de precisión está plagada de trabajos que proponen tratamientos más eficientes a partir de mapas de variabilidad. Por ejemplo, (Weisz, Fleischer, & Smilowitz, 1996), (Dobermann, y otros, 2002) y (Gerhards, Sökefeld, Schulze-Lohne, Mortensen, & Kühbauch, 1997) presentan aplicaciones selectivas de insecticida, nutrientes y herbicidas, respectivamente. Sin embargo, hasta ahora ningún trabajo ha llevado a la práctica la propuesta conjunta, es decir, la implementación de un sistema que permita lanzar misiones, registrar los datos que producen y reutilizarlos en otras misiones. Un sistema de estas características contribuiría a automatizar por completo la ejecución de las tareas agrícolas mediante flotas de robots aéreos y terrestres, de ahí que esta integración sea uno de los objetivos principales de la presente tesis.

Para completar todavía más este sistema, se propone que no solo contenga la lógica necesaria para ejecutar, coordinar y combinar misiones agrícolas de diferente naturaleza, sino que se integre dentro de un sistema global de gestión de misiones, que permita planificar, controlar y supervisar misiones, analizar y procesar los datos que generan, almacenarlos en una base de datos, combinarlos con otros sistemas para incrementar el conocimiento que se tiene del cultivo, etc.

En este sentido, sí que existen aplicaciones software para analizar datos extraídos de cultivos, pero se limitan al procesamiento de la información (Optimizer, 2015)(FarmLogs, 2015) o incluso a definir las rutas de trabajo manualmente (Farm Works™ Mapping, 2015), pero sin realizar un control en tiempo real sobre la flota que suministra estos datos.

3 PLANIFICACIÓN DE TAREAS AGRÍCOLAS

Este capítulo trata el problema de encontrar los planes –los cuales incluyen las rutas– que deben seguir los vehículos de actuación terrestres para llevar a cabo conjuntamente y de forma óptima una tarea agrícola. El problema se aborda desde un punto de vista muy general, teniendo en cuenta multitud de aspectos que pueden influir en los planes y rutas asociadas, a saber, las características de la flota (número de vehículos, velocidades de trabajo, radios de giro, capacidad de los tanques, ancho de trabajo, etc.), el campo de cultivo (forma, número de líneas, dirección de cultivo, distribución de las malas hierbas, etc.), el tipo de tarea (pulverización, inspección, tarea con cobertura total o parcial, etc.), y el criterio de optimización (distancia recorrida, coste por insumo, tiempo invertido, etc.).

La estructura del capítulo es la siguiente. La primera sección formula el problema de planificación en términos de una optimización combinatoria. La segunda se centra en los algoritmos de optimización que se han empleado para resolver el problema. La tercera detalla los procesos de configuración mediante los cuales se han ajustado los algoritmos utilizados. Finalmente, en la última sección, se evalúan los algoritmos comparándolos entre sí y resolviendo un amplio y variado repertorio de problemas que sintetizan muchos de los escenarios habituales en planificación agrícola; analizando, a la vista de los resultados, tanto la validez de los métodos empleados como la validez del planteamiento general adoptado.

3.1. Formulación: problema de optimización combinatoria

En agricultura los campos se siembran habitualmente en filas contiguas de tal modo que, cuando el cultivo crece, están formados por líneas paralelas de plantas. Entre cada dos líneas adyacentes queda un pequeño surco o espacio vacío, habitualmente denominado calle, que recorre longitudinalmente el campo y cuya anchura viene determinada por las necesidades de agua, luz o espacio de la especie cultivada. En ocasiones este espacio es lo suficientemente ancho (cultivos de calle ancha, véase Tabla 1.2) como para permitir la navegación dentro del cultivo sin tocar las plantas, o haciéndolo de forma muy leve e inofensiva, por ejemplo, circulando el vehículo al completo por dentro de las calles (robots muy pequeños o calles muy anchas, véase

Figura 3.1a y Figura 3.1b) o situando solo las ruedas en su interior (superando el bajo del vehículo el cultivo gracias a la altura de éstas, véase Figura 3.1c). No obstante, en ambos escenarios los vehículos están obligados a moverse siguiendo siempre la dirección de cultivo, esto es, la dirección en la que están dispuestas las plantas, sin cambiar las ruedas de calle, ya que, de hacerlo, se pisarían las líneas y se dañarían seriamente las plantas y, además, los surcos provocarían muchas vibraciones que afectarían al correcto funcionamiento de la maquinaria, fundamentalmente de los aperos.

Cuando el espacio entre líneas es muy pequeño, por ejemplo, en cultivos de cereal (cultivos de calle estrecha, véase Tabla 1.2), resulta imposible navegar por el interior sin pisar las plantas y lo que se hace es limitar la navegación a etapas muy tempranas de crecimiento, cuando los tallos son muy flexibles y mucho más tolerantes al paso de las ruedas (Figura 3.1d). En este caso, al igual que sucede con los escenarios anteriores, los surcos aunque menos pronunciados, también producen vibraciones, por lo que es conveniente seguir la dirección de cultivo para garantizar el buen funcionamiento de la maquinaria y más aún cuando se utilicen aperos de precisión. En definitiva, una vez que se ha establecido el cultivo, tanto si el espacio entre líneas es pequeño o grande, como si el estadio de crecimiento de las plantas es temprano o tardío, la maquinaria siempre debe moverse en la dirección de cultivo cuando trabaja en el interior del campo.



(a)



(b)



(c)



(d)

Figura 3.1. Tipos de navegación dentro de cultivo. (a) Vehículo pequeño dentro de cultivo de calle estrecha, (b) vehículo medio dentro de cultivo de calle ancha, (c) vehículo medio con solo ruedas dentro de calles de cultivo de calle ancha y (d) vehículo medio desplazándose en cultivo de calle estrecha en etapa temprana de crecimiento.

Otra restricción importante es la de evitar la circulación marcha atrás, puesto que la mayoría de los aperos agrícolas no están preparados para trabajar en ese sentido. Por ejemplo, así ocurre con arados, gradas, etc. Incluso en el caso de aquellos aperos que sí podrían por estar suspendidos (barras de pulverización, atomizadores, etc.), si se quisiera, por ejemplo, rectificar la trazada a la salida del campo maniobrando hacia atrás para conectar mejor con el siguiente punto de entrada, habría una alta probabilidad de dañar las líneas de cultivo y, además, el movimiento extra acentuaría la compactación del suelo de la zona. Por tanto, en este trabajo se parte de una situación real en la que las maniobras siempre se ejecutan fuera de la zona de cultivo, más concretamente en las cabeceras, que son las zonas que se sitúan en los extremos del cultivo con objeto de permitir el giro y desplazamiento de los vehículos, y con ello el tránsito entre las distintas zonas del cultivo.

Por otro lado, las tareas deben realizarse sin dejar zonas sin atender y de forma disjunta, es decir, garantizando que no se realiza la misma tarea dos veces en la misma zona (sin solapamientos). Atendiendo a estos requisitos la representación de un cultivo que se utiliza en este trabajo de investigación divide el campo en franjas disjuntas (a las que se llamará pistas de ahora en adelante por constituir las vías de tránsito dentro del cultivo), paralelas y en la dirección de cultivo, con una anchura igual al ancho de trabajo de los aperos que transportan los vehículos. El objetivo es recorrer todas las pistas exactamente una vez mediante alguno de los vehículos de la flota.

Como no es posible girar dentro de las pistas, las rutas serán secuencias de pistas intercaladas con las maniobras para transitar entre ellas. La Figura 3.2 muestra un campo con 7 pistas, todas ellas orientadas por construcción según la dirección de cultivo y formadas por 5 calles (4 interiores más media a cada lado), porque el ancho de trabajo de la herramienta empleada, en el ejemplo, abarca 5 líneas de cultivo.

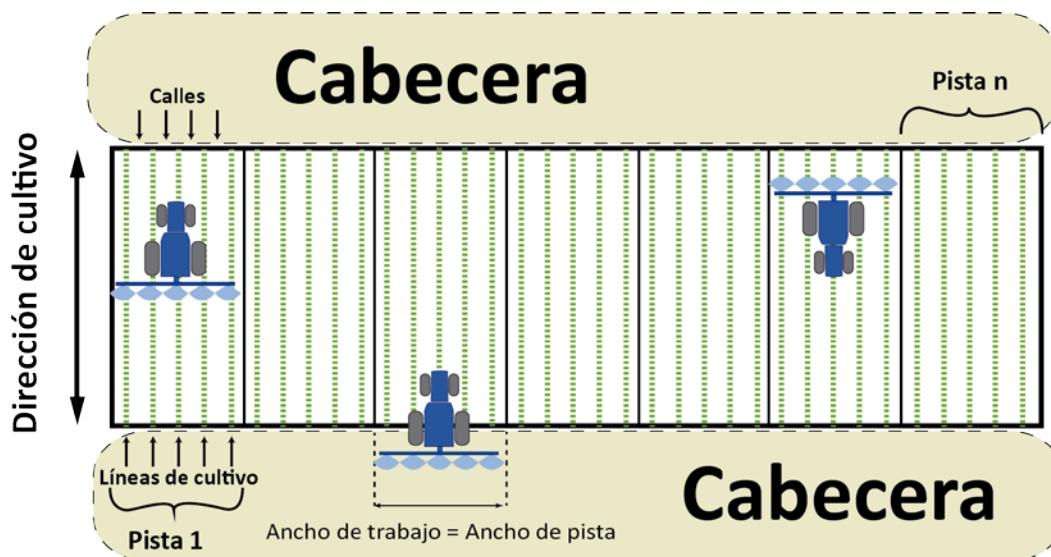


Figura 3.2. Campo dividido en 7 pistas de 5 líneas de cultivo cada una.

Es importante destacar la generalidad de la representación propuesta, ya que permite representar campos de forma irregular y de dirección variable –como los que se muestran en la Figura 3.3– usando pistas de diferentes longitudes y con distintas formas para adaptarlas a la dirección de cultivo.



Figura 3.3. (a) Campos con formas irregulares y con direcciones de cultivo variables, y (b) su representación en pistas.

Matemáticamente, el problema puede ser entonces visto como el de encontrar la secuencia de pistas que minimice el coste global de la tarea, ya que el orden en el que se recorren las pistas determina las trayectorias a seguir y, a su vez, las trayectorias inciden en el coste global. Se trata de un problema de optimización combinatoria, en el que hay que encontrar la secuencia óptima dentro de un conjunto finito de combinaciones posibles.

Como se tienen que recorrer todas las pistas exactamente una vez, con un solo vehículo el problema podría formularse como un problema clásico de optimización combinatoria, más exactamente como un TSP según las ecuaciones (3.1) y (3.2), pero sin el requisito de la vuelta a la pista de partida.

$$\sigma^* = \arg \min_{\sigma} f(\sigma) \quad (3.1)$$

$$f(\sigma) = \sum_{i=1}^{|\sigma|-1} c_{\sigma(i)\sigma(i+1)} \quad (3.2)$$

Siendo σ una permutación de las pistas a visitar (como en la secuencia de pistas no puede repetirse ninguna pista, puede expresarse mediante una permutación); $\sigma(i)$ la pista en la posición i ; $f(\sigma)$ la función que mide el coste que implica la solución σ ; σ^* la permutación óptima, es decir, aquella que minimiza el coste; $|\sigma|$ el número de pistas a visitar y c_{ij} el coste de transitar de la pista i a la pista j .

Sin embargo, en el presente trabajo de investigación el problema tiene que abordarse de una forma más general para adaptar la aproximación al trabajo de una flota de vehículos, es decir, que el problema a resolver sería entonces similar al mTSP. En términos matemáticos el problema se podría expresar con una ecuación similar a (3.1), pero incluyendo elementos separadores en la solución para distinguir el conjunto de pistas que debe visitar cada vehículo. Las soluciones x estarían ahora compuestas por la permutación σ de las pistas y los separadores s_i según las ecuaciones (3.3), (3.4) y (8):

$$x^* = \arg \min_x f(x) \quad (3.3)$$

$$f(x) = g(\sigma(1:s_1)) + \sum_{i=1}^{m-2} g(\sigma(s_i + 1:s_{i+1})) + g(\sigma(s_{m-1} + 1:n)) \quad (3.4)$$

$$g(\sigma) = \sum_{i=1}^{|\sigma|-1} c_{\sigma(i)\sigma(i+1)} \quad (3.5)$$

Siendo x^* la solución óptima, $f(x)$ la función que mide el coste de la solución x , $\sigma(i:j)$ la sub-permutación o parte entre i y j de la permutación σ , m el número de vehículos en la flota, n el número de pistas, s_i el separador que separa las pistas en σ del vehículo i de las del vehículo $i + 1$ y $g(\sigma)$ la función que evalúa el coste de la permutación o sub-permutación σ de modo similar a la ecuación (3.2).

Además se tiene que cumplir la condición (3.6) sobre los separadores de forma que estos estén acotados entre 0 y n , siendo n ($= |\sigma|$) el número total de pistas, y que, para cualesquiera dos de los $m - 1$ primeros vehículos i, j del conjunto V con todos los vehículos de la flota, el separador s_i asociado al vehículo i tiene que indicar una posición menor o igual en σ que la del separador s_j asociado al vehículo j . Para el último vehículo de la flota, el vehículo m ($= |V|$), no es necesario separador ya que se asume que le corresponden las pistas a partir del último separador s_{m-1} hasta la parte final de la permutación σ .

$$0 \leq s_i \leq s_j \leq n, \quad \forall i, j \in V, 1 \leq i \leq j \leq m - 1 \quad (3.6)$$

Con la nueva formulación es posible obtener aquellas rutas que realizan una cobertura total del campo sin solapamientos en un coste mínimo, sin embargo, este modelo todavía no contempla la capacidad de carga de los vehículos para almacenar/transportar los insumos. Considerando esta limitación, en términos matemáticos, el problema es similar al CVRP y puede expresarse añadiendo la restricción de capacidad, ecuación (3.7), a las ecuaciones (3.3), (3.4) y (3.5).

$$\sum_{i=1}^{|\sigma_k|} d_{\sigma_k(i)} \leq D \quad (3.7)$$

Siendo σ_k la sub-permutación de σ con las pistas del vehículo k –esto es, $\sigma(1:s_i)$ para el vehículo 1, $\sigma(s_{m-1} + 1:n)$ para el vehículo m y $\sigma(s_i + 1:s_{i+1})$ para el resto de vehículos–, d_i la cantidad necesaria de insumo para trabajar la pista i y D la cantidad de insumo que puede transportar un vehículo de la flota.

Sin embargo, hay una diferencia muy relevante respecto al problema CVRP que hace que el problema agrícola real sea más complejo. En el CVRP los vehículos tienen una determinada capacidad para almacenar mercancías que no puede ser recargada en ningún caso, de tal modo que una de las precondiciones del problema es que la suma total de las capacidades de los vehículos sea mayor o igual que la suma de todas las demandas de los clientes, ya que en caso contrario no existiría solución. Si se asumiera esta condición en el caso agrícola, solo se podría trabajar con campos que tuviesen una necesidad de insumos menor o igual a la inicialmente transportada por el conjunto de la flota, es decir, no existiría la posibilidad de reabastecimiento. Claramente esta

simplificación limita la generalidad del problema formulado ya que independientemente del tamaño de la flota y de la capacidad de los vehículos es imposible garantizar que se pueda realizar cualquier tarea en cualquier campo sin poder hacer reabastecimientos. En otras palabras, el reabastecimiento es una operación inherente a las tareas agrícolas y como tal debe ser considerado en la formulación del problema.

Extender el problema clásico CVRP para considerar los reabastecimientos aumenta considerablemente la dificultad de formulación del mismo. Una primera aproximación podría suponer que basta con acudir al depósito de suministros cada vez que se acabe una pista y el vehículo no disponga de suficiente insumo para tratar la siguiente pista, sin embargo, ésta no tiene por qué ser la mejor estrategia. El momento óptimo de reabastecimiento puede variar dependiendo de la localización del depósito de suministros, la cantidad de insumo restante en el vehículo y la cantidad requerida para terminar la tarea. Por ejemplo, si el depósito está cerca al terminar de trabajar una pista, puede ser interesante realizar un reabastecimiento incluso teniendo suministro suficiente para tratar las próximas pistas si de este modo se puede evitar el viaje de reabastecimiento desde un punto más alejado del campo. Por otro lado, puede suceder que incluso teniendo poco insumo no sea necesario acudir al depósito si se redistribuyen las pistas entre los vehículos de la flota de tal modo que el vehículo afectado tenga suficiente con la cantidad que transporta.

Como las pistas se tratan de forma individual, los reabastecimientos solo pueden realizarse durante las transiciones entre una pista y la siguiente, así que el sub-problema de reabastecimiento se reduce a encontrar las transiciones óptimas para ir y volver del depósito. Matemáticamente el problema se puede expresar con las ecuaciones anteriores (3.3)-(3.7), pero ahora la solución x estará compuesta por una tupla (σ, s, b) donde σ es la permutación de pistas, s el vector de separadores, y b es un vector binario con tantas componentes como transiciones entre pistas hay en σ y donde una componente b_i es igual a 0 si no hay reabastecimiento durante la transición i , e igual a 1 en caso contrario, es decir, cuando el vehículo debe acudir al depósito de suministros nada más abandonar la pista tratada a reabastecerse antes de continuar su tarea con la pista siguiente. Teniendo en cuenta esto, la restricción de capacidad formulada en la ecuación (3.7) no se aplica ahora a cada σ_k sino que debe cumplirse para cada una de sus partes entre reabastecimientos, por ello, la ecuación (3.7) debe reformularse como se muestra en las ecuaciones (3.8) y (3.9).

$$\sum_{i=1}^{|\pi|} d_{\pi(i)} \leq D \quad (3.8)$$

$$\pi \in \{\sigma_k(p:q) \mid b(\sigma_k(l)) = 0, 1 \leq p \leq l < q \leq |\sigma_k|\} \cup \{\sigma_k(l), 1 \leq l \leq |\sigma_k|\} \quad (3.9)$$

Donde π es cada parte entre reabastecimientos de la sub-permutación σ_k , es decir, son partes $\sigma_k(p:q)$ donde o no hay ningún reabastecimiento o solo hay uno y es al salir de la última pista, $\sigma_k(q)$.

Considerar reabastecimientos es una extensión que aumenta significativamente la complejidad del problema, ya que si antes la optimización tenía $m - 1 + n$ variables de decisión, siendo m el número de vehículos de la flota (de ahí que sea necesario determinar $m - 1$ separadores) y n el número de pistas del campo, añadiendo el vector de reabastecimientos se añaden a las anteriores $n - m_u$ variables binarias de decisión, porque hay $n - m_u$ transiciones entre pistas, siendo $m_u (\leq m)$ el número de vehículos

usados en la solución, y por tanto $n - m_u$ puntos en los que hay que determinar si se reabastece o no. Como el número de pistas (n) suele ser generalmente bastante superior al número de vehículos, prácticamente se duplica el número de variables en un problema que es NP-completo.

Otra situación real y que puede ser frecuente es que los vehículos en la flota no tengan todos las mismas características, es decir, que la flota sea heterogénea. Contemplar este caso supone modificar la ecuación (3.5) para considerar los costes asociados de cada vehículo, tal como recoge la ecuación (3.10), donde c_{ij}^k es el coste de transitar de la pista i a la j usando el vehículo k . Asimismo, también hay que actualizar la restricción de capacidad formulada en la ecuación (3.8) para que tenga en cuenta la capacidad particular asociada a cada vehículo, tal como se muestra en la ecuación (3.11).

$$g(\sigma) = \sum_{i=1}^{|\sigma|-1} c_{\sigma(i)\sigma(i+1)}^k \quad (3.10)$$

$$\sum_{i=1}^{|\pi|} d_{\pi(i)} \leq D_k \quad (3.11)$$

Cuando los vehículos son homogéneos, en términos de coste es indiferente para la optimización el vehículo concreto con el que se trate cada pista, puesto que todos los vehículos realizarán la misma tarea con el mismo coste asociado. De ahí que en ese caso la optimización esté únicamente en la distancia recorrida en las transiciones para conectar las pistas. Cuando se consideran vehículos con diferentes características no es así porque ya no es indiferente el vehículo que se emplea en cada pista. Por ejemplo, un vehículo con un menor consumo de combustible será siempre preferible a uno que consuma más. Para poder considerar esta heterogeneidad en los vehículos de la flota cuando se planifica, es necesario actualizar la función de coste $g(\sigma)$ para que incluyan los gastos asociados de tratar cada pista. Así, partiendo de la ecuación (3.10), se obtendría la ecuación (3.12), donde se incluyen los costes c_i^k de tratar la pista i con el vehículo k .

$$g(\sigma) = c_{\sigma(|\sigma|)}^k + \sum_{i=1}^{|\sigma|-1} (c_{\sigma(i)\sigma(i+1)}^k + c_{\sigma(i)}^k) \quad (3.12)$$

Por último, la reformulación propuesta permite plantear el problema como una multi-optimización que considere varios criterios simultáneamente. Para ello hay que sustituir la ecuación (3.3) por la ecuación (3.13):

$$X^* = \arg \min_x (f_1(x), \dots, f_l(x)) \quad (3.13)$$

Siendo x una posible solución, l el número de objetivos/criterios distintos a minimizar, $(f_1(x), \dots, f_l(x))$ el vector de los costes objetivo a minimizar y X^* el conjunto de soluciones óptimas (ya que en este tipo de optimizaciones puede haber varias, como se verá más adelante).

Este tipo de problemas multi-objetivo se resuelven con métodos capaces de establecer relaciones de orden entre soluciones con distintos costes (Deb, Multi-objective Optimization, 2014).

En resumen, el problema de planificación para llevar a cabo una tarea agrícola de forma óptima se aborda bajo una amplia perspectiva, obteniendo por construcción soluciones con cobertura total y sin solapamientos que consideran: 1) múltiples vehículos con distintas características, 2) las características de la propia tarea y 3) múltiples objetivos o criterios de optimización. Asimismo, con la formulación del problema propuesta no importa la forma del campo ya que la actuación se descompone en un conjunto de pistas con el ancho de tratamiento y que siguen el trazado de las líneas de cultivo, en definitiva el problema se reduce, en todos los casos, a encontrar el mejor orden para recorrer todas las pistas con independencia de donde estén situadas o el trazado que tengan. Es importante también observar que las tareas que requieren una cobertura parcial (por ejemplo, un tratamiento de control de plagas en el que solo se desee acudir a las pistas infestadas) son un sub-problema de las que emplean cobertura total (en vez de considerar todas las pistas se considerarían solo las infestadas), y que por tanto también se abarca este caso mediante la formulación propuesta.

3.2. Algoritmos meta-heurísticos

La práctica más habitual de resolver un problema de optimización combinatoria es utilizar un algoritmo meta-heurístico (Blum & Roli, 2003). Los algoritmos meta-heurísticos se aplican en aquellos problemas computacionales, habitualmente con espacios de búsqueda muy grandes, en los que no se conoce ningún método que obtenga siempre la solución óptima. El nombre de estos algoritmos proviene del empleo de mecanismos heurísticos dentro de una estructura de funcionamiento genérica y abstracta, lo que permite que puedan adaptarse para abordar una amplia variedad de problemas.

Existen multitud de algoritmos meta-heurísticos (véase una clasificación en la Figura 3.4). Los hay inspirados en la naturaleza, basados en poblaciones (trabajan con varias soluciones a la vez), guiados por funciones objetivo dinámicas, que implementan búsquedas locales, etc.

Todos estos algoritmos comparten propiedades y filosofía. Así son algoritmos no deterministas, o lo que es lo mismo, la misma entrada no tiene por qué producir siempre la misma salida. De hecho, son algoritmos de aproximación, porque construyen soluciones aproximadas al óptimo, generalmente acercándose a él de forma paulatina. Además son métodos generales, aplicables a problemas muy diferentes, que buscan las soluciones, de entre todas las existentes en el espacio de búsqueda, cuyas variables de decisión minimizan (o maximizan, dependiendo del problema) una función objetivo. Todos los métodos están estructurados y guiados por una meta-heurística aunque incorporan también un cierto grado de aleatoriedad en el mecanismo de búsqueda que les imprime el carácter indeterminista. Comparados con los algoritmos voraces, en los que la búsqueda está totalmente determinada por una heurística que conduce a un óptimo sin garantía de que sea el óptimo global, los algoritmos meta-heurísticos incluyen en su indeterminismo la clave para huir de los mínimos/máximos locales.

La complejidad del mecanismo de búsqueda en los métodos meta-heurísticos es polinomial, pero la aleatoriedad ralentiza la convergencia a una solución que, por lo general, está más próxima al óptimo que la solución suministrada por un algoritmo voraz.

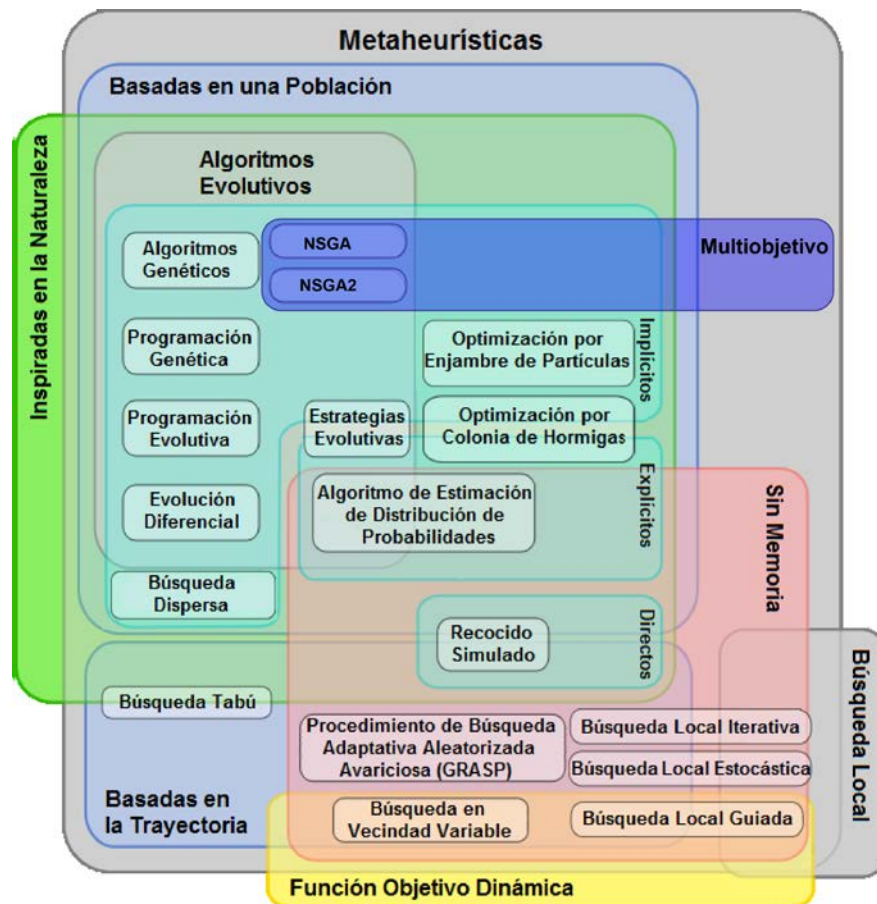


Figura 3.4. Clasificación de los algoritmos meta-heurísticos más habituales. La clasificación es una extensión de la propuesta en (Dréo, 2015) que incluye la categoría multi-objetivo

A groso modo los pasos que tiene que ejecutar un algoritmo meta-heurístico son: 1) generar un conjunto de hipótesis o posibles soluciones, 2) evaluar las soluciones que se han generado en términos de la cercanía al óptimo, 3) quedarse con las mejores soluciones y repetir el proceso mientras no se cumpla algún criterio de terminación, como por ejemplo, superar un periodo máximo de tiempo o alcanzar una solución suficientemente buena o cercana al óptimo.

Hay dos aspectos que son importantes a la hora de utilizar con éxito un método meta-heurístico porque directamente influyen en la convergencia o proceso iterativo de acercamiento hacia el óptimo: la forma en la que se generan las soluciones y el método de evaluación de la capacidad de una solución para resolver el problema, en otras palabras, lo alejada que está del óptimo.

Para generar las nuevas soluciones, se utilizan operadores que, según la meta-heurística empleada, pueden construir las soluciones inicialmente (sin información previa) o a partir de soluciones obtenidas en el ciclo previo. En ambos casos las soluciones se construyen incluyendo aleatoriedad en el proceso, con el fin de introducir un cierto grado de indeterminismo en la exploración. Los operadores que se utilizan para

construir soluciones iniciales se llaman generadores, mientras que los que utilizan la información de las soluciones seleccionadas en el ciclo previo se conocen como constructores de estados vecinos o mutadores (por la influencia de las meta-heurísticas inspiradas en estrategias evolutivas, tradicionalmente muy empleadas). Además, las nuevas soluciones se conocen como soluciones vecinas, estados vecinos o soluciones hijas (de nuevo, por influencia de las meta-heurísticas evolutivas). Estos operadores son una pieza clave porque implementan los saltos (también llamados transiciones) de exploración por el espacio de búsqueda. Tienen que permitir que cualquier punto del espacio de búsqueda sea alcanzable bien a través de una única transición o a través varias consecutivas. Las estrategias que utilizan estos operadores son variadas, por ejemplo, algunos construyen las nuevas soluciones a partir de una única solución, mientras que otros son capaces de hacerlo a partir de varias. Una vez generadas, el siguiente paso es evaluarlas, para ello se utiliza la función objetivo (función de *fitness* en las meta-heurísticas evolutivas). Esta función se encarga de estimar la aptitud de las soluciones en términos del criterio de optimización, otorgándoles un valor de aptitud en función de lo alejada que está la solución del óptimo. Gracias a la función de fitness se puede establecer una relación de orden entre todas las soluciones del espacio de búsqueda, lo que permite dirigir la búsqueda hacia unas regiones u otras según la aptitud de las soluciones analizadas en cada iteración del algoritmo.

Por último, el proceso de transición entre soluciones del espacio de búsqueda, se repite hasta que se cumple algún criterio de terminación, habitualmente, que la solución esté suficientemente próxima al óptimo, desconocido a priori y del que solo hay una estimación. De hecho, algunas técnicas meta-heurísticas llamadas exactas, si no se interrumpen pueden llegar a examinar todas las soluciones del espacio de búsqueda (en función del tipo de problema) garantizando la localización de un óptimo global. No obstante, el caso habitual es que los métodos meta-heurísticos sean probabilísticos, esto es, que solo den una garantía estadística de encontrar el óptimo (que aumenta con el tiempo de ejecución) tendiendo a examinar todos los candidatos y garantizar el óptimo al 100% solo para un tiempo infinito de ejecución.

De todos los algoritmos meta-heurísticos disponibles (véase Figura 3.4) en este trabajo de investigación se probaron tres: *Recocido Simulado*, *Algoritmos Genéticos* y *Non-Dominated Sorting Genetic Algorithm 2 (NSGA-II)*. Se seleccionaron estos algoritmos por ser métodos muy utilizados en la resolución de problemas clásicos de planificación de rutas, como los mencionados TSP, mTSP y CVRP.

3.2.1. Estrategia general de los algoritmos de optimización empleados

Los tres métodos propuestos comparten una estrategia común: 1) parten de un conjunto de soluciones X (de ahora en adelante también llamado conjunto de trabajo) generado aleatoriamente, 2) evalúan X para determinar si alguno de los criterios de terminación se cumple, como, por ejemplo, si alguna solución es lo suficientemente buena o si el tiempo máximo de búsqueda se ha excedido. En caso afirmativo la búsqueda se da por concluida y en caso contrario 3) las mejores soluciones en X se seleccionan y se guardan en un nuevo conjunto intermedio Y , 4) se construyen nuevas soluciones a partir de las contenidas en Y y se almacenan en X' (conjunto vecindario, conjunto de soluciones hijas o simplemente conjunto hijo), 5) se sustituye el conjunto de trabajo X por alguna

combinación de los conjuntos X y X' , y 6) el proceso se repite desde 2). Este funcionamiento se recoge en la Figura 3.5.

```

1   X = inicializarConjuntoDeSoluciones();
2   WHILE (NOT seCumpleAlgunaCondicionDeTerminacion(X))
3       Y = seleccionarMejoresSoluciones(X);
4       X' = construirNuevasSoluciones(Y);
5       X = combinarAntiguasYNuevasSoluciones(X, X');
6   END

```

Figura 3.5. Procedimiento que resume la parte más externa de funcionamiento de los tres algoritmos meta-heurísticos seleccionados

Los aspectos distintos entre los diferentes métodos seleccionados son: 1) El número de soluciones del conjunto de trabajo X es diferente según el método, por ejemplo, el Recocido Simulado trabaja con una única solución mientras que el Algoritmo Genético y el NSGA-II son métodos de población por lo que analizan simultáneamente más de una solución en cada iteración. Además, entre el Algoritmo Genético y el NSGA-II, la selección también es distinta. El primero solo escoge en función de un criterio, mientras que el segundo es un método de optimización multi-objetivo, por lo que usa varios. 2) La etapa de generación de las nuevas soluciones (véase línea 4 en Figura 3.5), en el Algoritmo Genético y en NSGA-II está dividida en dos etapas (cruce y mutación), mientras que en el Recocido Simulado solo existe un paso (generación del vecindario). 3) Por último, la fase de combinación y sustitución (véase línea 5 en Figura 3.5) es también distinta para los tres métodos.

En las siguientes secciones se comentan los detalles importantes en común para la implementación de los tres algoritmos en la resolución del problema propuesto, tales como la representación de la solución, los operadores para la construcción de nuevas soluciones y las funciones objetivo. Las diferencias y la estructura interna propia de cada método se explicaran más adelante.

3.2.2. Representación de la solución

Como se explicó en el apartado de formulación, toda solución al problema de planificación agrícola debe contener la información necesaria para determinar cómo se distribuyen las pistas entre los vehículos de la flota, en qué orden debe recorrer cada vehículo las pistas asignadas y si deben reabastecerse y, en caso afirmativo, entre qué pistas.

El orden de las pistas y su distribución entre los vehículos de la flota se puede representar mediante una permutación de la unión del conjunto P , que contiene todas las pistas p_i en las que se dividió el campo, y del conjunto S , que contiene $m - 1$ separadores s_i , siendo m el número de vehículos en la flota. De esta manera, independientemente de donde se sitúen los separadores, la permutación siempre estará dividida en m partes, siendo la primera parte la que contiene las pistas a cargo del primer vehículo; la segunda, las del segundo, y así sucesivamente. El orden en el que se deben recorrer las pistas lo determina el orden en el que aparecen en cada parte, de modo que se empieza por la pista

más a la izquierda y se continúa por la siguiente a la derecha hasta llegar a la última pista (la pista más a la derecha). Si alguna parte no contiene pistas (por contener la permutación dos separadores seguidos), significa que el vehículo correspondiente no interviene en la planificación.

Para los reabastecimientos, como se indicó en la formulación, bastará con un vector binario b con tantos elementos como transiciones haya en la permutación, indicando el valor 1 que hay reabastecimiento durante la transición i , y el valor 0 que no. Sin embargo, con esta representación, el vector debería tener un tamaño variable puesto que el número de transiciones depende del número de vehículos usados en la solución, en concreto el tamaño sería $n - m_u$, siendo n el número de pistas y m_u el número de vehículos usados. Para no usar representaciones variables, más difíciles de implementar, se decidió usar un vector con tantas componentes como pistas, redefiniendo el significado de cada b_i de tal modo que un 1 significara que se debe acudir a reabastecer entre la pista p_i y la siguiente (p_{i+1}), y 0 que no hay reabastecimiento. En el caso de que no existiera pista siguiente, por ser p_i la última de la ruta, b_i debe ignorarse independientemente de su valor por no tener sentido realizar un reabastecimiento tras haber terminado la tarea. Con esta representación el problema tiene $2n + m_u - 1$ variables decisión, siendo n de ellas binarias.

En lo que sigue, nos referiremos a la primera parte de la solución como vector permutación, vector de pistas o simplemente permutación, mientras que la segunda parte de la solución será el vector binario de reabastecimientos o simplemente vector de reabastecimientos.

Es importante destacar que con esta representación las soluciones por construcción no pueden dar lugar a colisiones entre los vehículos dentro del cultivo, ya que los vehículos trabajan siempre en pistas diferentes y disjuntas, sin embargo estas sí pueden darse en las cabeceras ya que es un lugar de tránsito común a todos los vehículos. Por esta razón se supone que la flota dispone de algún sistema para evitar las colisiones en cabecera. Como veremos en el siguiente capítulo (Capítulo 4) en este trabajo de investigación se propone un sistema para supervisar a los vehículos mientras llevan a cabo la tarea, que está constantemente monitorizando las trayectorias de todos los elementos de la flota y que es capaz de prevenir las colisiones potenciales pausando y reanudando a los vehículos involucrados.

3.2.2.1. Decodificación: obteniendo las trayectorias reales y el plan de actuación

Una vez identificadas las pistas a cargo de cada vehículo, en qué orden se deben recorrer y en qué puntos hay reabastecimientos, se dispone de toda la información necesaria para obtener las trayectorias exactas de cada vehículo. Dentro del campo las trayectorias están definidas por construcción a ser las líneas interiores centrales de cada pista que conectan los extremos, para que de ese modo el apero pase –y trabaje– por toda la pista.

El cálculo de las trayectorias asociadas a las transiciones es más complejo. Las transiciones pueden ser directamente entre dos pistas o haciendo escala en el depósito para reabastecer. En este segundo caso pueden representarse como una transición de ida hacia el depósito más una de vuelta. Como la posición del depósito y los puntos extremos de las pistas son bien conocidos, se sabe la posición de los puntos de partida y llegada para todas las transiciones. Sin embargo, como los vehículos reales no pueden girar

libremente en cualquier dirección porque tienen un radio mínimo de giro relacionado con sus limitaciones mecánicas, no es posible unir ambos puntos simplemente mediante una recta sino que es necesario saber las orientaciones de partida y llegada. Para el caso de los puntos en las pistas es fácil determinar las orientaciones de salida/entrada a partir de la dirección de cultivo, de si el punto es de salida o entrada, y de si está en la cabecera superior o en la inferior. Las ecuaciones (3.14) y (3.15) obtienen las orientaciones de salida θ_{out} y entrada θ_{in} para un punto p a partir de la dirección de cultivo θ_c , suponiendo que ésta se haya calculado en el sentido cabecera inferior - cabecera superior (véase sentido de θ_c en Figura 3.6).

$$\theta_{out}(p) = \begin{cases} \theta_c, & \text{si } p \text{ está en la cabecera superior} \\ -\theta_c, & \text{si } p \text{ está en la cabecera inferior} \end{cases} \quad (3.14)$$

$$\theta_{in}(p) = \begin{cases} -\theta_c, & \text{si } p \text{ está en la cabecera superior} \\ \theta_c, & \text{si } p \text{ está en la cabecera inferior} \end{cases} \quad (3.15)$$

Para el caso del depósito, la orientación de entrada y salida se suponen similares a las de la carretera de entrada/salida que da acceso/salida al recinto. En la Figura 3.6 se muestran las orientaciones de entrada y salida de las pistas de un campo y del depósito de suministros.

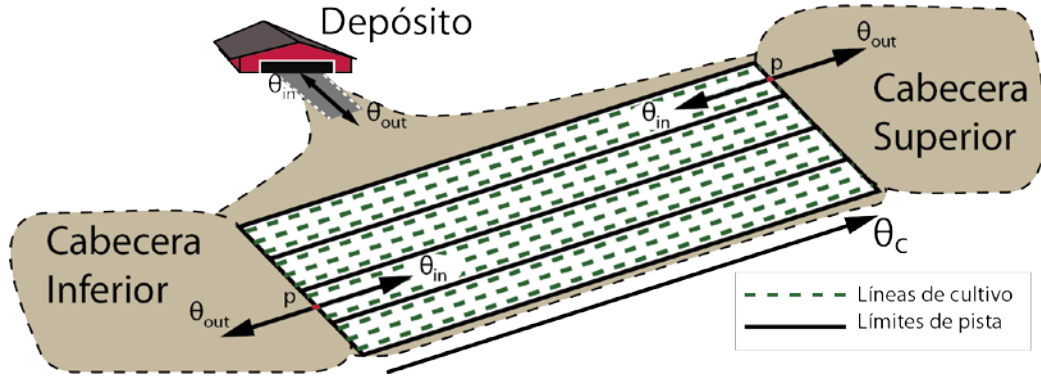


Figura 3.6. Orientaciones de un cultivo

Conocidas entonces todas las posiciones y orientaciones de salida y de entrada, el problema de encontrar las transiciones con distancia mínima se traduce a encontrar el camino más corto que debe seguir un vehículo con un radio de giro para llegar desde un punto de salida con una orientación inicial hasta un punto destino con una orientación determinada. Según el teorema de Dubins (Dubins, 1957), el camino más corto entre una configuración de salida (punto y orientación de salida) y una configuración de entrada/llegada (punto y una orientación de llegada) es siempre una secuencia de líneas rectas (R) y arcos circulares (C) de radio r_{min} de la forma CRC, CCC o una subsecuencia de estas dos últimas, siendo r_{min} el mínimo radio de giro posible. Extrapolando a trayectorias de vehículos, como éstos pueden girar hacia la izquierda o la derecha, los dos tipos de trayectorias posibles se pueden desglosar en 6 casos que surgen de particularizar los arcos circulares (C) en giros a la izquierda (I) o la derecha (D). Los 6 casos resultantes son: IRI, IRD, DRI, DRD, IDI y DID (véase Figura 3.7). En otras

palabras, cualquier trayectoria mínima entre una configuración de entrada y otra de salida siempre tendrá una forma que encaje en alguno de estos 6 casos (o en sus sub-casos).

Para saber cuál de los casos contiene la trayectoria mínima dadas dos configuraciones cualesquiera, una estrategia habitual consiste en evaluar los 6 casos y quedarse con el que da lugar a la ruta más corta, aunque (Shkel & Lumelsky, 2001) demostraron que mediante razonamientos geométricos es posible saber de antemano cual es el caso que contiene la trayectoria mínima, ahorrando así mucho tiempo de cómputo, lo que es de gran interés si hay que planificar en tiempo real o se tienen que evaluar muchas trayectorias. Esta última situación es a la que nos enfrentamos en este trabajo de investigación, ya que los algoritmos meta-heurísticos funcionan construyendo multitud de soluciones que tienen que ser evaluadas.

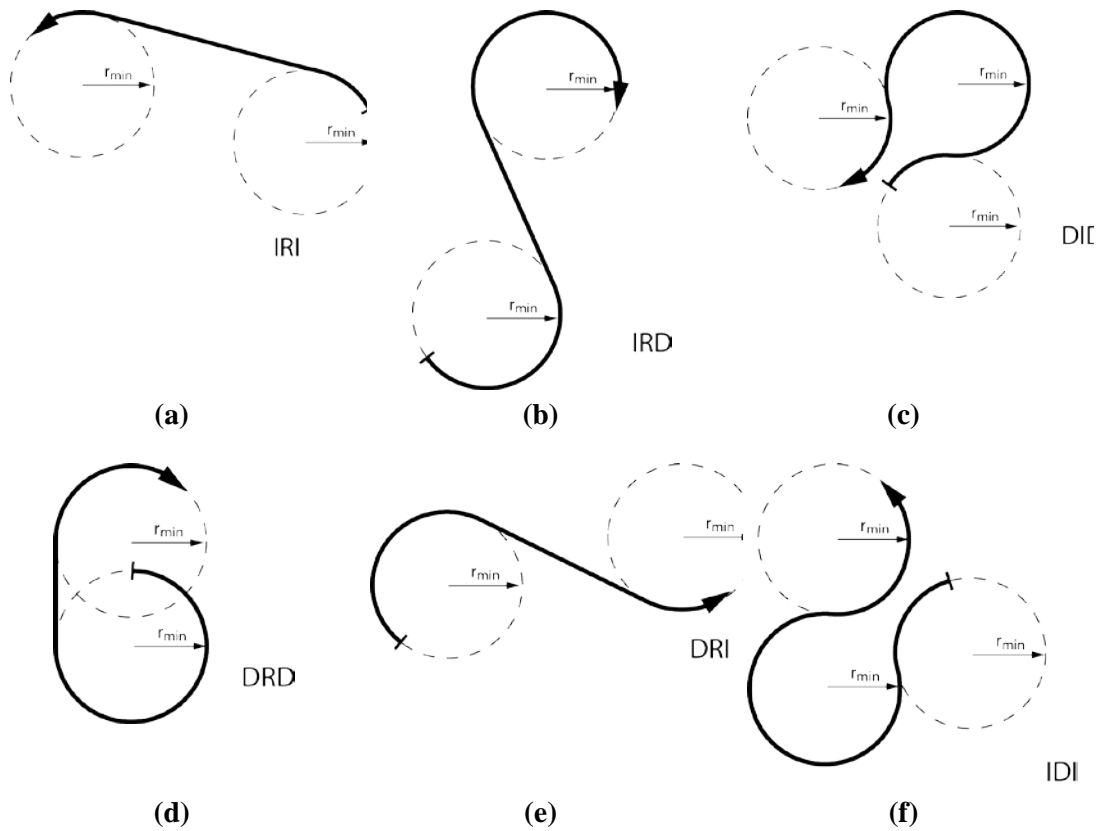


Figura 3.7. Las 6 trayectorias posibles según el teorema de Dubins (Dubins, 1957) entre dos puntos con orientación de salida y orientación de entrada fijas.

Por tanto, mediante (Dubins, 1957) y (Shkel & Lumelsky, 2001) es posible conocer el camino más corto para cualquier transición. No obstante, como se mostró en la Figura 2.1, estos giros se pueden concretar en maniobras Π y Ω –que se modelan con las ecuaciones (2.1)-(2.3)– cuando se trabaja en campos rectangulares y se transita entre pistas. Para los casos en los que es necesario trabajar con transiciones más generales, como por ejemplo entre pistas y depósitos o entre pistas en campos no rectangulares, se puede utilizar el procedimiento más general descrito en (Shkel & Lumelsky, 2001).

Una vez conocido como hallar las trayectorias mínimas para las transiciones, ya sea conectando directamente las pistas, ya sea intercalando una visita al depósito, basta con conectarlas con las trayectorias intra-pista para formar la ruta de cada vehículo.

La Figura 3.8 muestra, a modo de ejemplo, las trayectorias codificadas en la solución formada por la permutación $\sigma = (p_3, p_6, p_8, p_{10}, s_2, p_7, p_2, p_1, s_1, s_3, p_9, p_4, p_5)$ y el vector binario de reabastecimientos $b = (1, 0, 1, 0, 0, 0, 0, 0, 0)$. El campo está compuesto por 10 pistas ($|P| = 10$), cada una de ellas formada a su vez por 4 calles. El campo contiene rodales de malas hierbas (rectángulos verdes en la Figura 3.8) que deben ser tratados por los vehículos mediante pulverización de herbicida, para lo que disponen de barras de pulverización de 4 m con 4 boquillas, una para cada calle. La flota cuenta con 4 vehículos, es decir, que $|S| = 3$, aunque el tercer vehículo no forma parte de la solución porque en σ no hay ninguna pista a su cargo. Inicialmente se sabe si los vehículos parten de la cabecera superior o inferior, por lo que es fácil inferir el extremo de pista por el que tienen que comenzar. El ancho de giro de los vehículos es 3 m, lo que permite el tránsito entre pistas adyacentes mediante maniobras Ω , como se muestra en la transición entre la primera y la segunda pista en la Figura 3.8.

El vector de reabastecimientos indica que los vehículos que tratan la primera y la tercera pista del campo deben acudir al depósito antes de comenzar a tratar las pistas siguientes. Sin embargo, puesto que la primera pista no tiene siguiente (es la última pista que trata el vehículo 2, véase la Figura 3.8), en ese caso no es necesario acudir a reabastecer.

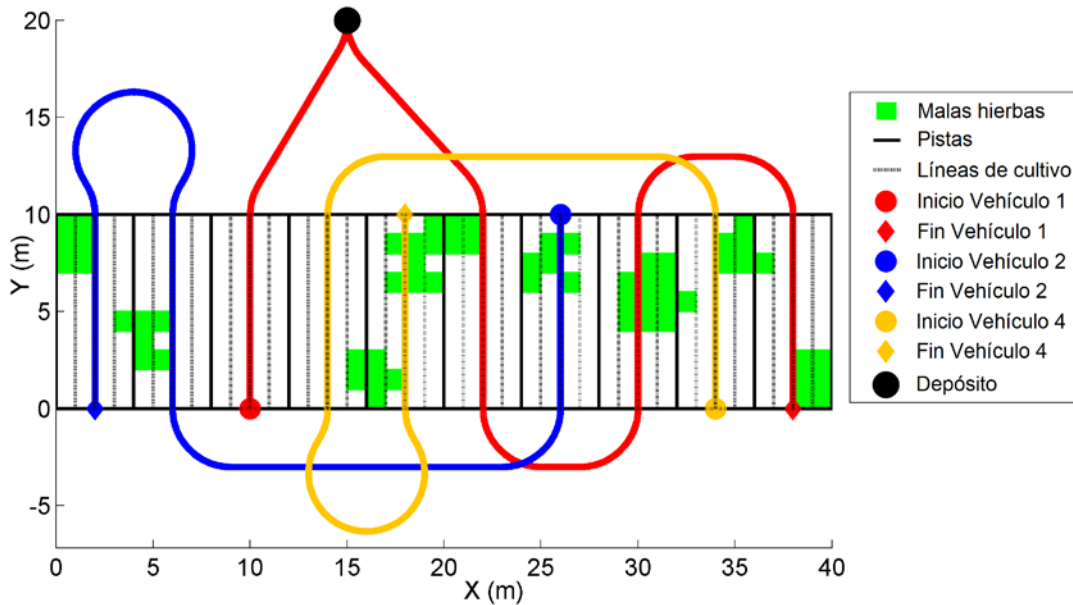


Figura 3.8. Trayectorias asociadas a la permutación $\sigma = (p_3, p_6, p_8, p_{10}, s_2, p_7, p_2, p_1, s_1, s_3, p_9, p_4, p_5)$ y el vector de reabastecimientos $b = (1, 0, 1, 0, 0, 0, 0, 0, 0)$

Además de las trayectorias se puede extraer automáticamente de la solución el plan de actuación de la tarea a realizar, esto es, el estado (velocidad, apero activado o desactivado) que deben tener los vehículos en cada uno de los puntos de las rutas. Para

ello hay que confeccionar una lista de los puntos y orientaciones que definen las trayectorias (puntos y orientaciones de entrada y salida de las pistas o el depósito) junto con sus estados asociados, más los puntos y orientaciones intermedios en los que se produce un cambio en el estado que afecta a la ejecución de la tarea, en otras palabras, un cambio de velocidad o un cambio en el estado del apero (cambio en el estado de las boquillas). Para calcular el estado del apero en cada punto, además de las trayectorias, es necesario usar la información adicional que se dispone del campo y de la tarea, en este caso el mapa de malas hierbas que indica donde están los rodales. A partir de las trayectorias, la longitud de la barra, la distribución de las boquillas en la barra, y la posición de los rodales es fácil calcular en qué puntos deben abrirse y cerrarse las boquillas para pulverizar solo las malas hierbas. La Tabla 3.1 muestra, a modo de ejemplo, el plan del vehículo 1 de la solución de la Figura 3.8.

Tabla 3.1. Plan del vehículo 1 codificado en la solución de la Figura 3.8

Vehículo	Paso	Posición	Orientación	Velocidad (km/h)	Boquillas de la barra de pulverización				Descripción
					V1	V2	V3	V4	
1	1	(10, 0)	90°	6	OFF	OFF	OFF	OFF	Punto inicial
1	2	(10, 10)	90°	10	OFF	OFF	OFF	OFF	Salida pista 3
1	3	(15, 20)	90°	10	OFF	OFF	OFF	OFF	Llegada a depósito
1	4	(15, 20)	270°	10	OFF	OFF	OFF	OFF	Salida de depósito
1	5	(22, 10)	270°	6	OFF	OFF	ON	ON	Inicio pista 6
1	6	(22, 8)	270°	6	OFF	OFF	OFF	OFF	Cambio en boquillas
1	7	(22, 0)	270°	10	OFF	OFF	OFF	OFF	Fin pista 6
1	8	(30, 0)	90°	6	OFF	OFF	OFF	OFF	Inicio pista 8
1	9	(30, 4)	90°	6	OFF	ON	ON	ON	Cambio en boquillas
1	10	(30, 7)	90°	6	OFF	OFF	ON	ON	Cambio en boquillas
1	11	(30, 8)	90°	6	OFF	OFF	OFF	OFF	Cambio en boquillas
1	12	(30, 10)	90°	10	OFF	OFF	OFF	OFF	Fin pista 8
1	13	(38, 10)	270°	6	OFF	OFF	OFF	OFF	Inicio pista 10
1	14	(38, 8)	270°	6	OFF	OFF	OFF	ON	Cambio en boquillas
1	15	(38, 7)	270°	6	OFF	OFF	OFF	OFF	Cambio en boquillas
1	16	(38, 3)	270°	6	ON	ON	OFF	OFF	Cambio en boquillas
1	17	(38, 0)	270°	0	OFF	OFF	OFF	OFF	Punto final

Para cada paso el vehículo tiene que adoptar el estado asociado y mantenerlo hasta el próximo paso. Por ejemplo, en los pasos relacionados con la octava pista del campo (pasos 8 a 12), se indica que la unidad debe partir del punto (30, 0) con una orientación de 90° a una velocidad de 6 km/h y con todas las boquillas de la barra apagadas hasta llegar al punto (30, 4) con la misma velocidad y orientación, allí se deben activar las boquillas V2, V3 y V4 (las tres más a la derecha) para empezar a pulverizar el rodal que hay en medio de la pista. Todo se mantendrá igual hasta el punto (30, 7) donde se desactivará la

boquilla V2 porque el rodal se reduce en anchura por su izquierda y, un metro más adelante en el punto (30, 8), se desactivarán las otras dos válvulas, la V3 y la V4. Por último, se mantiene todo igual hasta el final de la pista, el punto (30, 10), donde se cambiará la velocidad a 10 km/h (en cabecera no hay requisitos sobre la velocidad, por lo que se puede aumentar) para transitar hasta la siguiente pista, pista 10, donde, de nuevo, se puede apreciar las ordenes de apertura y cierre de las boquillas para tratar las malas hierbas que aparecen en esta pista.

Por último, en este ejemplo se ha supuesto que las posiciones de partida de los vehículos estaban en los extremos de sus respectivas pistas iniciales. Es importante resaltar que el procedimiento es el mismo si los puntos iniciales están fuera del campo (como la localización de un garaje, un depósito o cualquier punto que se precise), con la única salvedad de que hay que añadir a las trayectorias de cada vehículo la transición inicial necesaria para llegar desde el punto inicial con la orientación inicial hasta el punto de entrada de la primera pista con la orientación requerida. Del mismo modo cuando los puntos finales están fuera del campo habrá que añadir a la trayectoria las transiciones finales desde los puntos de salida del campo hasta los puntos finales definidos

3.2.2.2. Solución inicial

Debido a que los vehículos tienen un tanque limitado, no es posible generar totalmente al azar las soluciones iniciales de las que parten los algoritmos de optimización empleados, ya que podría suceder que un vehículo no tuviera capacidad suficiente para tratar todas las pistas que se le han asignado. Por ello se desarrolló un pequeño algoritmo para generar las soluciones iniciales. El algoritmo empieza por el primer vehículo de la flota y va asignando aleatoriamente pistas de entre las pistas no asignadas hasta que no queda suficiente contenido en el tanque del vehículo que permita tratar una nueva pista. En caso de quedar pistas por cubrir, el algoritmo pasa al segundo vehículo y repite el proceso, y así sucesivamente hasta acabar con todos los vehículos o todas las pistas. En caso de acabar con todos los vehículos y quedar pistas por tratar, se añade aleatoriamente un reabastecimiento final a alguno de los vehículos, para que vuelva a tener el tanque lleno, y se le asigna de nuevo aleatoriamente todas las pistas posibles. El proceso se repite hasta que no quedan pistas por asignar.

3.2.2.3. Reparación de soluciones no viables

Debido a las modificaciones aleatorias que los operadores de las meta-heurísticas realizan en las soluciones durante la etapa de generación, como se verá más adelante, puede suceder que una solución deje de ser viable por no disponer alguno de sus vehículos de suministro suficiente en su tanque para completar las pistas asignadas. Esto se puede solucionar enviando al vehículo a reabastecimiento en alguna de las transiciones previas a que se agote el contenido del tanque y posterior o igual a la primera transición en la que haya espacio suficiente para recargar la cantidad que faltará más adelante, ya que no tiene sentido parar a reabastecer si solo se puede cargar una cantidad inferior a la que se necesita. En caso de que haya varias transiciones en las que sea posible recargar, se selecciona una al azar y se marca como transición en la que hay acudir a depósito en el vector de reabastecimientos. Este es el proceso al que se ha llamado reparación de las soluciones no viables.

Reparar las soluciones no viables justo después de crearlas implica un coste computacional extra pero garantiza que siempre se trabaje con soluciones válidas, lo cual puede ser beneficioso para la exploración del espacio de soluciones, por lo que es necesario conocer esta alternativa. En la futura sección de configuración (apartado 3.3.2), se estudia si su uso genera resultados mejores que justifiquen el coste computacional extra que supone.

3.2.3. Funciones objetivo

Tanto en la etapa de evaluación de la condición de terminación (línea 2 de Figura 3.5) como en la de selección (línea 3), se deben evaluar las soluciones del conjunto de trabajo; en el primer caso para saber si alguna solución cumple alguna de las condiciones relacionadas con el fin de la búsqueda, y en el segundo caso, etapa de selección, para escoger las mejores soluciones. Estas evaluaciones se realizan mediante las funciones objetivo. Estas funciones calculan la calidad de las soluciones en términos de algún criterio (objetivo) de optimización mediante un valor (habitualmente llamado valor objetivo) que sirve de medida para saber cómo de buena es la solución y para compararla con otras soluciones.

En la sección 3.1 ya se mostró la forma genérica de estas funciones objetivo. En lo que sigue se consideran varios objetivos o criterios de optimización, tales como la distancia recorrida, el coste por insumos y el tiempo requerido o tiempo necesario para llevar a cabo la tarea agrícola. El coste por insumos se calcula a partir de la cantidad de combustible y herbicida utilizados, ya que la propuesta, aunque general, se focaliza al tratamiento de malas hierbas.

En general, cuando se trabaja con una flota, muchos de los costes de la tarea global se expresan como una combinación de los costes relativos a cada uno de elementos de la flota. En el caso de la distancia recorrida y el coste por insumos, el coste se expresa como la suma de todos los costes realizados por cada uno de los vehículos de la flota, mientras que el tiempo empleado en realizar la tarea se calcula como el máximo de los tiempos invertidos por cada uno de los vehículos de la flota, puesto que trabajan en paralelo y se supone que todos comienzan a la vez. Siendo, por tanto, el vehículo que invierte más tiempo el que determina el tiempo requerido para la realización de la tarea. Atendiendo a estas consideraciones, las ecuaciones (3.16), (3.17) y (3.18) expresan los costes en distancia $f_d(x)$, coste por insumos $f_I(x)$ y tiempo $f_t(x)$ respectivamente para una solución x en función de los costes parciales $g_d(x_i)$, $g_I(x_i)$ y $g_t(x_i)$. Donde los costes parciales son los realizados por cada vehículo $v_i \in V$ para completar su parte x_i de la solución x .

$$f_d(x) = \sum_{i=1}^{|V|} g_d(x_i) \quad (3.16)$$

$$f_I(x) = \sum_{i=1}^{|V|} g_I(x_i) \quad (3.17)$$

$$f_t(x) = \max_{1 \leq i \leq |V|} g_t(x_i) \quad (3.18)$$

Asimismo, la formulación de los costes de cada vehículo $g_d(x_i)$, $g_l(x_i)$ y $g_t(x_i)$ puede unificarse mediante tres ecuaciones con la misma apariencia lo que simplifica la posterior implementación. La idea consiste en llevar un registro de actividad con todas las operaciones realizadas por los vehículos que den lugar a algún tipo de gasto, por ejemplo, desplazamientos, arranques, reabastecimientos, etc., ya que, conociendo la actividad llevada a cabo y lo que cuesta cada operación individual, es posible calcular el coste que le lleva a cada vehículo realizar su parte de la tarea, tal como se muestra en las ecuaciones (3.19), (23) y (3.21).

$$g_d(x_i) = \sum_{j=1}^{|O|} (m_j \cdot c_{ij}^d) \quad (3.19)$$

$$g_l(x_i) = \sum_{j=1}^{|O|} (m_j \cdot c_{ij}^l) \quad (3.20)$$

$$g_t(x_i) = \sum_{j=1}^{|O|} (m_j \cdot c_{ij}^t) \quad (3.21)$$

Donde m_j es el número de operaciones de tipo $o_j \in O$ necesarias para llevar a cabo la parte x_i de la solución x . Mientras c_{ij}^d , c_{ij}^l y c_{ij}^t son los costes de la operación o_j en términos de distancia, coste por insumos y tiempo, respectivamente, cuando la ejecuta el vehículo v_i .

Esta forma de cálculo facilita la incorporación de nuevos objetivos de optimización y la implementación del cálculo de costes, ya que es posible asignar un registro de actividad a cada vehículo que recoja las distintas operaciones individuales realizadas por el vehículo y calcular el coste de la tarea realizada por el vehículo multiplicando las distintas operaciones individuales por el coste que tienen para el vehículo.

La aproximación propuesta, por tanto, requiere descomponer la tarea agrícola en el conjunto de operaciones elementales que supongan un coste en términos de cualquiera de los criterios de optimización definidos. A partir de todo lo expuesto, en este trabajo de tesis se han considerado las siguientes operaciones elementales $o_j \in O$:

- Arranque: consiste en poner en marcha un vehículo para usarlo en la tarea agrícola. El coste asociado está relacionado con el consumo de combustible que produce esta operación y el tiempo requerido en el caso de que haya una fase de inicialización de dispositivos, como por ejemplo la calibración de los sensores o el comienzo de recepción de la señal GPS. Asimismo, en el caso de utilizar vehículos alquilados, emplear más vehículos supone un mayor gasto, y esta posibilidad también puede expresarse incluyendo el coste de alquiler de cada unidad como gasto de arranque. Por último, el número de operaciones de arranque que aparecen codificadas en una solución es igual al número de vehículos empleados para realizar la tarea.
- Desplazamiento intra-pista: cada vez que se recorre un metro dentro del cultivo se realiza un desplazamiento intra-pista que consume combustible y tiempo y por tanto

debe estar en el registro de operaciones. La suma de estas operaciones corresponde, por tanto, con la distancia en metros recorrida dentro del cultivo.

- Desplazamiento inter-pista: Son similares a los desplazamientos intra-pista pero fuera del cultivo, en la cabecera, es decir, son los desplazamientos para transitar entre pistas, ya sea pasando o no por el depósito para realizar una recarga. Al igual que el caso del desplazamiento intra-pista, este desplazamiento produce consumo de combustible y tiempo por lo que tiene que ser considerado en el cálculo de los costes globales de una tarea. En este trabajo se distingue entre desplazamientos intra e inter-pista porque el movimiento de la maquinaria es distinto dentro y fuera del cultivo. Por ejemplo, la velocidad dentro del cultivo viene determinada por el tipo de tarea (siembra, tratamiento de pulverización, etc.) mientras que en el exterior la velocidad depende de aspectos como el terreno, el tipo de vehículo o las normas de seguridad.
- Operación de actuación: es la operación propia de la tarea que se lleva a cabo. Por ejemplo, si la tarea es la inspección del campo, la actuación podría consistir en tomar fotos y geo-referenciarlas, lo que no tiene un coste realmente significativo. Sin embargo, no sucede lo mismo cuando la actuación se encamina al tratamiento de algún tipo de plaga utilizando un fitosanitario; en ese caso, dependiendo de la dosis, el coste de la actuación puede ser más o menos elevado, razón por la cual este tipo de coste debe incluirse en el registro de operaciones.
- Reabastecimiento o repostaje: aprovisionamiento (combustible, herbicida, etc.) que se realiza cuando es necesario visitar un depósito de suministros. Su coste no depende de la cantidad de insumo almacenado en el vehículo, puesto que éste ya se contabiliza mediante las operaciones de actuación o, en el caso del combustible, mediante los desplazamientos inter-pista. Tampoco contabiliza las distancias recorridas fuera del cultivo para ir y volver del depósito, ya que se registran también mediante las operaciones de desplazamiento inter-pista. El coste para este tipo de operación se estima en función del tiempo requerido para realizar el repostaje.

Una vez conocidas las operaciones que se deben incluir en el registro, es fácil contabilizarlas a partir de la solución. El número de arranques es el número de vehículos que se usan en la solución (aquellos que tienen al menos una pista asignada), los desplazamientos intra- e inter-pista corresponden directamente con la distancia recorrida en el cultivo y en cabeceras, y pueden ser calculados a partir de las trayectorias. El número de reabastecimientos puede extraerse fácilmente del vector de reabastecimientos. Por último, el número de operaciones de actuación puede extraerse a partir de las trayectorias, el ancho de trabajo del apero y la información adicional que se tiene del campo de cultivo, del mismo modo que se calculan los estados del apero para el plan de actuación explicado en el apartado 3.2.2.1.

Para agilizar el cómputo de algunas de estas operaciones se utilizaron técnicas de memoización. La memoización (Michie, 1968) es una práctica habitual en optimización que consiste en almacenar los resultados de funciones costosas que deben invocarse en multitud de ocasiones exactamente con las mismas entradas, de este modo se evita perder tiempo calculando una y otra vez lo mismo. Concretamente, lo que se pre-calculó y almacenó en una tabla fueron las distancias para todas las posibles transiciones, agilizando, de este modo, el cómputo de los desplazamientos inter-pista.

Por último, como se ha explicado con anterioridad, en ocasiones es interesante optimizar varios objetivos simultáneamente, aunque de los tres algoritmos propuestos, solo NSGA-II es un método multi-objetivo. Para el resto de algoritmos es posible

extender los métodos para que traten con más de un objetivo definiendo una función objetivo que combine varios criterios, aunque esto conlleve algún tipo de transformación en los criterios para que todos se expresen en las mismas unidades. Por poner un ejemplo, si uno de los criterios es el tiempo que tardan los robots en realizar la tarea, el coste asociado podría calcularse en euros a partir del salario del operario a cargo de supervisar el funcionamiento de la flota

Con este enfoque lo que se optimiza es la combinación y no los objetivos que la componen, pero aun así es una estrategia interesante que puede resultar útil cuando lo que se quiere minimizar es, por ejemplo, la suma de varios objetivos. Siguiendo esta idea, la ecuación (3.22) puede usarse para minimizar, por ejemplo, la suma del coste por insumos y el tiempo, donde w_I y w_t son los pesos asignados a cada criterio y sirven para unificar las unidades.

$$f(x) = w_I \cdot f_I(x) + w_t \cdot f_t(x) \quad (3.22)$$

Por último, como por construcción las soluciones no evitan las colisiones en cabecera, se podría pensar en incluir algún tipo de penalización en el coste de aquellas soluciones cuyas trayectorias pudieran producir colisiones, sin embargo, este planteamiento implica un alto coste computacional, ya que habría que detectar para cada trayectoria puntos de gran proximidad para dos o más vehículos y aún así no se podría garantizar la ocurrencia de la colisión, ya que, como se comprobará experimentalmente en el Capítulo 4, es difícil que los vehículos agrícolas sigan gran exactitud las velocidades y las trayectorias especificadas en el plan de actuación por multitud de razones, entre ellas las condiciones del terreno, tiempo de respuesta de la maquinaria, etc. Ante esta situación la estrategia más adecuada consiste en incluir sensores y métodos en los robots de la flota de forma que estos puedan evitar obstáculos y disponer de un sistema de supervisión global de la flota, capaz de detectar conflictos entre los vehículos y evitarlos en tiempo real, tal como se verá en el Capítulo 4.

3.2.3.1. Costes de las operaciones

Para calcular los costes de cada una de las operaciones contenidas en el registro, es necesario estimar el gasto que originan para cada objetivo y expresarlo en unidades adecuadas. El gasto se puede calcular a partir de la cantidad de recursos que se utilizan (Tabla 3.2).

Tabla 3.2. Recursos utilizados por cada operación

Operación	Distancia	Tiempo	Combustible	Herbicida
Arranque		✓	✓	
Desplazamiento intra-pista (1 m)	✓	✓	✓	
Desplazamiento inter-pista (1 m)	✓	✓	✓	
Actuación (pulverización - 1 dosis)				✓
Reabastecimiento		✓		

Comenzando por la distancia, las únicas operaciones que implican movimiento son los desplazamientos que, por definición, se contabilizan con 1 metro. Los reabastecimientos no consumen distancia porque, como ya se comentó, la ida y la vuelta al depósito se contabilizan en desplazamientos intra-pista.

Respecto al tiempo, el arranque se puede estimar en unos 2 minutos, debido a la inicialización de los dispositivos internos. Los desplazamientos intra-pista dependen directamente de la velocidad a la que se mueven los vehículos, y ésta, a su vez, de la tarea agrícola que se esté realizando. Por ejemplo, según (Navas, y otros, 2010) el arado suele realizarse a 8 km/h, la rastra y la siembra a 12 km/h, la cosecha a 4 km/h, la pulverización de herbicida a 6 km/h, etc. Por tanto, si nos centramos en la pulverización de herbicida, que es la tarea de referencia a lo largo de esta memoria, el coste del desplazamiento intra-pista sería 0,6 s. En las cabeceras las velocidades pueden ser superiores, por ejemplo, de 10 km/h, por lo que el coste de los desplazamientos inter-pista podría estimarse en 0,36 s. El coste en tiempo de la operación de actuación no suele existir ya que, en general, la operación se realiza mientras el vehículo se desplaza y, por tanto, no retrasa en modo alguno la tarea. Finalmente, el reabastecimiento consume el tiempo de llenado del tanque que puede variar dependiendo del volumen del mismo. En esta tesis se ha considerado un valor medio de 2 minutos.

El coste por combustible de cada operación puede calcularse a partir del consumo medio del vehículo, que en general viene determinado por su potencia. En (Márquez, 2007) se habla de consumos de entre 2,32 y 8,16 €/h para tractores de entre 28 y 170 CV respectivamente. Como los tractores del proyecto RHEA son los modelos de New Holland Boomer 3050 de 50 CV de (Boomer 3000 series, 2015), se seleccionó esta potencia como valor de referencia. En concreto, según (Márquez, 2007), un tractor de 50 CV tiene un coste medio por combustible de 3,00 €/h. Como este coste se calculó sobre una base de 0,50 €/l, actualizando al precio del gasóleo B, que es el que consumen habitualmente los tractores, al valor que tenía el 15 Julio de 2014, 0,96 €/l (COAG, 2015), el consumo del tractor considerado sería 5,76 €/h. Además, como la velocidad de trabajo se ha fijado en 6 km/h y cada hora cuesta 5,76 € en gasóleo, el coste en combustible de las operaciones de desplazamiento, que por construcción implican moverse 1 m, puede estimarse en $0,96 \cdot 10^{-3}$ € que es el resultado de dividir los 5,76 € que cuesta la hora entre los 6.000 m que el tractor recorre en una hora.

Respecto al coste por arranque, para el estudio que se presenta a continuación se ha estimado el coste de la puesta en marcha de un vehículo en una cantidad simbólica distinta de 0, en concreto, 0,2 €. El objetivo es dejar preparado el método de planificación para una alternativa más realista que podría estimar el coste de arranque en función del coste de alquilar el vehículo, lo que se aproxima a una situación similar a la que encontraríamos si el tratamiento con una flota de tractores autónomos se facilita a los agricultores como un servicio, como propone RHEA, de forma similar al servicio de cosecha que ya se proporciona en la actualidad.

Sobre los costes por herbicida, tan solo la operación de pulverización implica gasto. Los tratamientos por herbicida pueden variar entre 10 €/ha, si se usan herbicidas baratos como el MCPA (MCPA 60 SL Herbicida, 2015), y 90 €/ha, usando productos más caros como el Rimsulfuron (Andújar, Ribeiro, Fernández-Quintanilla, & Dorado, 2011). Por tanto, si se supone una dosis por pulverización de 1 m^2 , el coste podría oscilar entre $0,1 \cdot 10^{-2}$ y $0,9 \cdot 10^{-2}$ €/por dosis.

Finalmente, si se considera el criterio coste por insumos, los costes para cada operación se obtienen de sumar los costes de todos los insumos consumidos. Esto es, combustible y herbicida en el escenario que se ha tomado como referencia en esta tesis.

La Tabla 3.3 resume los costes considerados por operación de modo que los costes se seleccionarán según el objetivo que se desee optimizar en cada momento. A excepción de la distancia, todos los costes dependen de las características del vehículo que se emplee, incluidos los dispositivos a bordo. Así, el equipo de actuación puede ser más o menos rápido, o ser más o menos eficiente energéticamente en la aplicación de tratamientos; por todo esto, los valores de la Tabla 3.3 deben entenderse como orientativos, aunque este hecho no influye en la solidez y generalidad del estudio que se presenta en las siguientes secciones.

Tabla 3.3. Costes estimados por operación y por criterio de optimización para tractores de 50 CV

Operación	Distancia (m)	Tiempo (s)	Combustible (€)	Herbicida (€)	Coste por insumos (€)
Arranque	0	120	0,2	0	0,2
Desplazamiento intra-pista (1 m)	1	0,6	$0,96 \cdot 10^{-3}$	0	$0,96 \cdot 10^{-3}$
Desplazamiento inter-pista (1 m)	1	0,36	$0,96 \cdot 10^{-3}$	0	$0,96 \cdot 10^{-3}$
Pulverización (1 dosis/ 1 m ²)	0	0	0	$0,1 \cdot 10^{-2}$ – $0,9 \cdot 10^{-2}$	$0,1 \cdot 10^{-2}$ – $0,9 \cdot 10^{-2}$
Reabastecimiento	0	120	0	0	0

3.2.3.2. Gap y cotas de los valores objetivo

Aunque las funciones objetivo permiten cuantificar numéricamente la bondad de las soluciones y establecer así una relación de orden entre ellas, dada una única solución, su valor objetivo asociado no da idea de cómo de próximo está del óptimo salvo que éste se conozca a priori. Por ello es habitual en optimización trabajar en términos de *gap* (Applegate, Bixby, Chvatal, & Cook, 2006) cuando se conoce el óptimo, o lo que es lo mismo calcular la diferencia entre el valor objetivo y el óptimo, en términos porcentuales del óptimo, de modo que cuánto más cerca esté de 0 mejor será la solución evaluada. Véase la ecuación (3.23), donde x es una solución cualquiera, x^* es una solución óptima, $f(x)$ es el valor objetivo asociado de la solución x y $f(x^*)$ es el valor óptimo asociado a la solución x^* .

$$gap(x) = \frac{f(x) - f(x^*)}{f(x^*)} \quad (3.23)$$

Los *gap* se utilizan sobre todo en problemas clásicos y de prueba, donde el óptimo se conoce de antemano. En los problemas reales donde precisamente lo que se está buscando es el óptimo, lo habitual es calcular una cota inferior del óptimo, un punto inalcanzable pero que indica el valor al que hay que tender en el límite. En algunas

ocasiones también puede ser interesante calcular una cota superior del peor valor posible, para de este modo estimar el tamaño del espacio de búsqueda.

En el caso de estudio, la cota inferior se puede estimar a partir del conjunto mínimo de operaciones necesarias para completar la tarea, por ejemplo, el número mínimo de arranques sería 1 porque al menos hay que utilizar un vehículo. Es importante que las estimaciones sean siempre una cota inferior de las operaciones individuales necesarias, de esta forma se garantiza que la cota inferior del óptimo sea siempre inferior a éste.

La Tabla 3.4 lista las operaciones indispensables que siempre se ejecutan como mínimo para completar una tarea. Los desplazamientos intra-pista son siempre, por construcción, un sumatorio de la longitud de cada una de las pistas que componen el cultivo, ya que cada pista debe recorrerse una y solo una vez. El mínimo número de desplazamientos inter-pista se puede estimar calculando la transición más corta posible, es decir, aquella que produzca el giro Π más corto con el radio de giro más pequeño de toda la flota, y el mínimo número de transiciones posibles que resulta de usar todos los vehículos. El número de pulverizaciones se puede estimar dividiendo el área infestada entre la superficie que cubre una dosis de herbicida. Por último, los reabastecimientos mínimos se estiman a partir de las pulverizaciones mínimas y el máximo número de dosis que se pueden almacenar entre todos los tanques de la flota.

Tabla 3.4. Número mínimo de operaciones que se ejecutan por tarea

Operación	Mínimo número de ejecuciones
Arranques	1
Desplazamientos intra-pista*	$\sum_{i=1}^{ P } longitud(p_i)$
Desplazamientos inter-pista*	$transiciones_{min} \cdot min_{transicion}$
Pulverizaciones	$A_{infestada}/A_{dosis}$
Reabastecimientos	$\left\lceil pulverizaciones_{min} / \sum_{i=1}^m tanque(v_i) \right\rceil - 1$

*Coincide con la distancia en metros

En la Tabla 3.4 $p_i \in P$ son las pistas del campo; $longitud(p_i)$, la longitud de la pista p_i ; $transiciones_{min}$, el número mínimo de transiciones; $min_{transicion}$, la longitud de la transición más corta posible; $pulverizaciones_{min}$, el mínimo número de pulverizaciones; $A_{infestada}$, el área infestada; A_{dosis} , el área que se trata con una dosis; m , el número de vehículos en la flota, y $tanque(v_i)$, la capacidad de tanque del vehículo v_i .

Conociendo las operaciones mínimas a realizar y usando los costes mínimos para cada operación en la flota considerada, es posible calcular las cotas inferiores de las funciones objetivo (3.16), (3.17) y (3.18), a partir de las ecuaciones (3.24), (3.25) y (3.26) respectivamente.

$$cota_{inferior}(f_d(x)) = \sum_{i=1}^{|O|} \min_{j=1:m} (c_{ij}^d) \cdot ops_{min}(o_i) \quad (3.24)$$

$$cota_{inferior}(f_I(x)) = \sum_{i=1}^{|O|} \min_{j=1:m} (c_{ij}^I) \cdot ops_{min}(o_i) \quad (3.25)$$

$$cota_{inferior}(f_t(x)) = \frac{\sum_{i=1}^k \min_{j=1:m} (c_{ij}^t) \cdot ops_{min}(o_i)}{m} \quad (3.26)$$

Siendo $ops_{min}(o_i)$ el mínimo número de operaciones necesarias de tipo o_i .

Como las operaciones contabilizadas son una cota inferior de las que realmente serían necesarias y los costes empleados para cada una de ellas son los mínimos disponibles entre los vehículos de la flota, el coste total real siempre será mayor que la cota calculada.

En el caso de que el criterio a minimizar sea el tiempo, como los vehículos trabajan en paralelo es necesario dividir por el número total de vehículos.

Para obtener la cota superior del peor valor objetivo posible, el procedimiento es análogo al anterior. Así se calcula una estimación del máximo número de operaciones posibles, es decir, aquellas que se deberían llevar a cabo para construir la peor solución posible, y después se calcula el valor de las funciones (3.16), (3.17) y (3.18) junto con los costes máximos por operación. En la Tabla 3.5 se muestran el número máximo de operaciones que se podrían realizar por tarea.

Tabla 3.5. Número máximo de operaciones que se ejecutan por tarea

Operaciones	Máximas ejecuciones
Arranques	m
Desplazamientos intra-pista	$\sum_{i=1}^{ P } longitud(p_i)$
Desplazamientos inter-pista	$maxdistancia_{reabastecimientos} + maxdistancia_{cabecera}$
Pulverizaciones	$A_{infestada}/A_{dosis}$
Reabastecimientos	$transiciones_{max}-1$

En la Tabla 3.5 m es el número de vehículos en la flota; $maxdistancia_{reabastecimientos}$, la máxima distancia que se puede recorrer en reabastecimientos; $maxdistancia_{cabecera}$, la máxima distancia que se puede recorrer en cabecera que resulta de recorrer el campo en el peor orden posible y usando el mayor radio de giro de entre todos los vehículos de la flota.

Conociendo el máximo número de operaciones a realizar y usando los costes máximos existentes en la flota para cada operación es posible calcular cotas superiores de las funciones objetivo según la ecuación (3.27):

$$cota_{superior}(f(x)) = \sum_{i=1}^{|O|} \max_{j=1:m} (c_{ij}) \cdot ops_{max}(o_i) \quad (3.27)$$

Siendo $f(x)$ cualquiera de las funciones (3.16), (3.17) y (3.18) y $ops_{max}(o_i)$ el máximo número de operaciones que puede realizarse de tipo o_i .

Como las operaciones contabilizadas son una cota superior de las que realmente serían necesarias y los costes empleados para cada una de ellas son los máximos disponibles entre los vehículos de la flota, el coste total real siempre será menor que la cota calculada.

3.2.4. Selección de las mejores soluciones

Una vez evaluadas las soluciones mediante las funciones objetivo, es posible elegir las mejores soluciones, aunque esta selección se realiza de distintas formas en función del método de optimización que se esté utilizando, por esa razón se abordará su explicación de forma individualizada para cada método.

3.2.5. Generación de las nuevas soluciones

Tras seleccionar las mejores soluciones, éstas se utilizan para construir un nuevo conjunto de soluciones con las que trabajar en la siguiente iteración, siendo esta la estrategia de exploración del espacio de búsqueda. Los operadores son los mecanismos que se utilizan para construir las nuevas soluciones. En este trabajo de investigación se han utilizado operadores de recombinación o cruce, que involucran a más de una solución, y de vecindad o mutación, que parten de una única solución. Por lo general a lo largo del texto se hablará de cruce y mutación, aunque en el caso del método de Recocido Simulado sería más apropiado usar el término vecindad en lugar de mutación.

A continuación se describen los operadores de cruce y mutación/vecindad empleados en esta tesis. Como la representación adoptada para la solución está formada por dos partes claramente diferenciadas, el vector permutación y el vector de reabastecimientos, se usarán diferentes operadores de cruce y mutación para cada una de estas partes.

3.2.5.1. Operadores de Cruce

Los operadores de cruce parten de un conjunto de soluciones (generalmente dos progenitores/padres) que recombinan para generar un conjunto de nuevas soluciones (descendientes/hijos). Se fundamentan en construir nuevas soluciones que combinan parte de las propiedades presentes en las soluciones de partida introduciendo cierto grado de aleatoriedad. En otras palabras, forman soluciones en regiones del espacio relativamente próximas a los progenitores, de modo que la zona es suficientemente amplia como para que no haya mucha endogamia y el proceso de convergencia avance, pero a la vez no excesivamente grande para que la búsqueda no se vuelva aleatoria.

Las técnicas que se usan para recombinar las soluciones dependen de la estructura de datos que se utilice para almacenarlas y de los tipos de valores que alberguen, pero cuando se emplean vectores de enteros, como sucede en la representación escogida en este trabajo, las técnicas más empleadas son: cruce en un punto, cruce en dos puntos y cruce uniforme (Holland, 1975). En estos tres tipos de cruce los progenitores son dos y el operador produce dos descendientes (véase Figura 3.9). El cruce en un punto divide ambas soluciones padre en dos partes, cortándolas por un mismo punto seleccionado de

forma aleatoria y recombinándolas, conectando la parte izquierda de la primera solución con la parte derecha de la segunda y viceversa. El cruce en un dos puntos divide las soluciones padre en tres partes, cortándolas por dos puntos escogidos de forma aleatoria, y las recombina intercambiando las partes centrales. Por último, el cruce uniforme usa un vector binario generado de forma aleatoria y de la misma longitud que los padres. Para aquellas componentes en las que el vector binario contenga un 1, se intercambian los valores de las mismas componentes en los padres, en caso contrario se dejan igual. Al ser una probabilidad uniforme, en media, un 50% de las componentes de las soluciones hijo provienen de componentes padre intercambiadas.

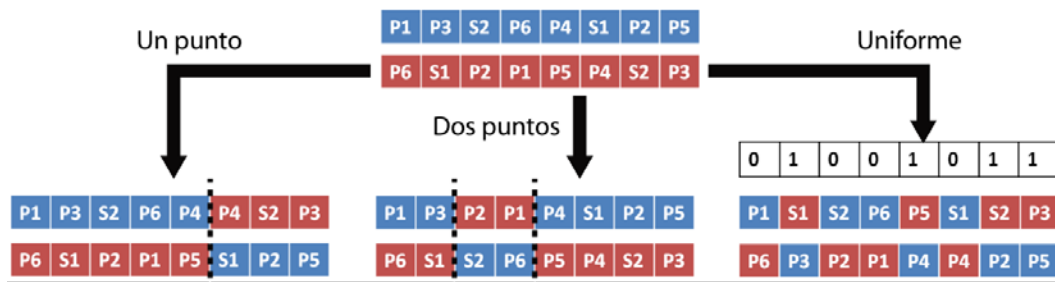


Figura 3.9. Cruce en (a) un punto, (b) dos puntos y (c) uniforme

Estas técnicas de recombinación son válidas para cruzar los vectores de reabastecimientos pero no los vectores permutación, ya que estos últimos representan permutaciones y, por tanto, no se puede intercambiar libremente las componentes del vector sin garantizar que las estructuras resultantes no contengan repeticiones y que sigan representando permutaciones.

En 1985 (Goldberg & Lingle, 1985) propusieron un cruce especialmente diseñado para permutaciones (PMX - *partially matched crossover*) y lo utilizaron para resolver el problema TSP mediante un Algoritmo Genético. El cruce PMX consiste en un cruce por dos puntos que tiene en cuenta la localización de las componentes de los vectores progenitores de modo que si se produce una repetición, se pueda intercambiar el elemento/componente repetido por otro que ocupa la misma posición. El algoritmo es el siguiente:

- 1) Se eligen aleatoriamente dos puntos de cruce.
- 2) Se intercambian los dos segmentos intermedios que permanecerán sin cambios hasta el final de la ejecución del operador. Hasta aquí, el operador se comporta igual que un cruce convencional en dos puntos.
- 3) Se evalúa el resto de los vectores (parte izquierda y parte derecha) para comprobar si existen elementos repetidos:
 - a) Si un elemento no está contenido en el segmento intercambiado, no existe repetición y el elemento permanece igual.
 - b) Si por el contrario está contenido, se intercambia por el elemento homólogo al repetido (aquel que ocupa la misma posición que el elemento repetido pero en el segmento intermedio de la otra solución). Si el nuevo elemento sigue estando repetido, se vuelve a aplicar el mismo paso, hasta que se encuentre un elemento no repetido.

La Figura 3.10 muestra gráficamente un cruce PMX. Tras aplicar un cruce por dos puntos para obtener el primer hijo se obtiene ($p_1, p_3 \mid p_2, p_1, p_5, p_4 \mid p_2, p_5$) donde p_1, p_2 y p_5 están repetidos (en negrita). Para cada uno de estos elementos se busca el homólogo al elemento repetido, respectivamente p_6, s_2 y p_4 . Se sustituyen p_1 y p_2 (en los segmentos extremos, no en el segmento central) por p_6 y s_2 (ya que ni p_6 ni s_2 están en el primer hijo). Asimismo, p_5 no se puede sustituir por p_4 porque seguiría estando repetido, por tanto, se repite el proceso pero esta vez para el nuevo valor p_4 , es decir, se busca el valor homólogo de p_4 en el primer progenitor, este valor es s_1 y como no aparece en el hijo se cambia por p_5 .

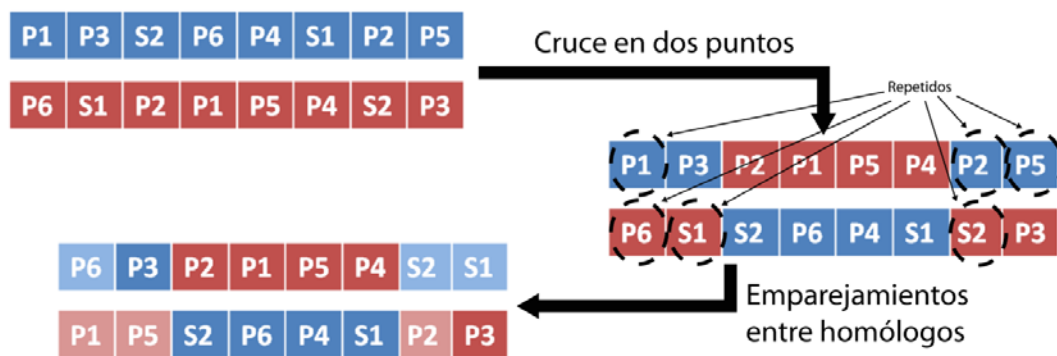


Figura 3.10. Cruce PMX. Primero se realiza un cruce en dos puntos y después emparejamientos entre homólogos hasta eliminar las repeticiones.

Respecto a la parte de la solución formada por el vector de reabastecimientos se ha optado por utilizar el cruce en dos puntos por ser sencillo y adaptarse bien a una representación binaria (De Jong, 2006), además de mostrar una buena tasa de mezcla (Prügel-Bennett, 2001).

3.2.5.2. Operadores de Mutación

A diferencia de los operadores de cruce, los de mutación parten de una única solución que modifican de forma aleatoria. Una modificación cuya finalidad es saltar de una zona a otra cualquiera del espacio de búsqueda por diferente que sea, ya sea en un único paso o en varios. En otras palabras, estos operadores permiten que la búsqueda no se estanque en una zona como puede suceder con el cruce si todas las soluciones son muy parecidas. De este modo se consigue un comportamiento de búsqueda más global o extensivo, que complementa al de los operadores de cruce, más local o intensivo (De Jong, 2006).

El primer operador de mutación específico para permutaciones que se probó fue el operador *scramble* (Syswerda, 1991). Este operador genera aleatoriamente una máscara binaria uniforme de la misma longitud que el vector a mutar y se permutan las componentes que corresponden con un 1 en la máscara. Como lo que se está haciendo es permutar elementos, el vector resultante sigue siendo una permutación, por lo que la técnica es aplicable tanto a los vectores permutación como a los vectores de reabastecimientos. Sin embargo, una vez utilizado el operador, se comprobó que el

rendimiento de los métodos de optimización en los que se empleó era bajo. La razón hay que buscarla en el número de 0's y 1's de la máscara binaria que en media es aproximadamente igual, lo que produce un gran número de cambios en cada aplicación del operador, obstaculizando la convergencia hacia mejores soluciones, especialmente en aquellas etapas de búsqueda delicadas, por ejemplo, en las cercanías del óptimo, en las que los cambios deben ser muy precisos y/o pequeños. Se intentó mejorar el rendimiento del operador usando distribuciones no uniformes con objeto de producir un menor número de 1's y por tanto de cambios, pero aún así, la convergencia era lenta respecto al uso de otros operadores de mutación. Se probaron entonces los operadores que se han usado tradicionalmente para resolver problemas similares al planteado en este trabajo de investigación como el TSP o el CVRP. Estos operadores son: *relocate* (Affenzeller, Wagner, Winkler, & Beham, 2009), *or-opt* (Or, 1976), *swap* (Oliver, Smith, & Holland, 1987), *simple inversion* (Holland, 1975), *cut inversion* (Larrañaga, Kuijpers, Murga, Inza, & Dizdarevic, 1999) y *LSM* (Or, 1976). El operador *relocate* elimina un elemento de la ruta y lo inserta en otra posición elegida aleatoriamente. El operador *swap* intercambia dos elementos entre sí. El operador *simple inversion* divide la ruta en tres partes, invierte el orden de la parte central y las reconecta. El operador *cut inversion* es similar al *simple inversion* pero insertando la parte invertida en otra posición al azar en lugar de en el mismo punto. El operador *or-opt* coge un grupo de elementos consecutivos y lo inserta en otra posición. Por último, el operador *LSM* (de *Local Search Mutation*) realiza una búsqueda local tomando un grupo de elementos consecutivos y probando a insertarlos en diferentes puntos, finalmente se queda con aquel que produce la mayor mejora en la ruta. A diferencia del operador *scramble*, que transforma demasiado las soluciones, los cambios que producen estos operadores favorecen la construcción incremental de las rutas óptimas porque realizan modificaciones relativamente pequeñas y en zonas concretas. La Figura 3.11 muestra gráficamente cada uno de estos operadores.

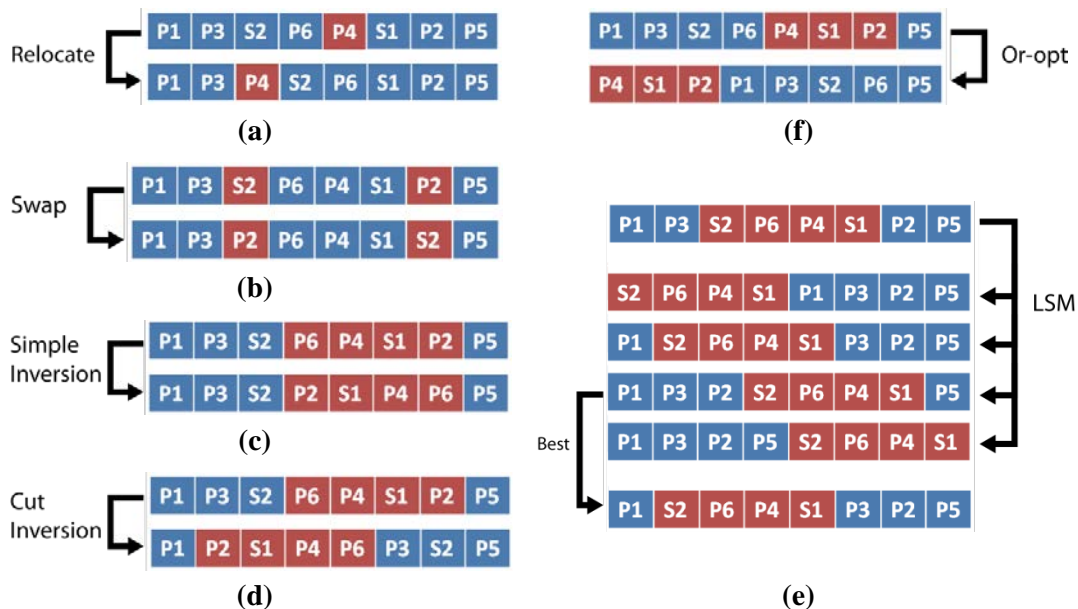


Figura 3.11. Operadores de mutación típicos en problemas de rutas. (a) *Relocate*, (b) *swap*, (c) *simple inversion*, (d) *cut inversion*, (f) *or-opt* y (e) *LSM*.

La mayoría de estos operadores se llevan usando muchos años por su simplicidad y buenos resultados y muchos de ellos son también conocidos con otros nombres, por ejemplo, el operador *relocate* también se le conoce por *insertion* (Michalewicz, 1996); el operador *or-opt* por *displacement* (Michalewicz, 1996); el operador *swap* por *exchange* (Banzhaf, 1990); el operador *simple inversion* por *2-opt* (Lin, 1965) y, finalmente, el operador *cut inversion* por *inversion* (Fogel, 1993). En esta memoria se ha optado por utilizar la nomenclatura más reciente.

La mayoría de estos operadores se diseñaron pensando en la resolución del problema TSP y, en consecuencia, para intercambiar, desplazar, insertar, etc. elementos de una única ruta. Para adaptar el funcionamiento de estos operadores a la resolución de un problema CVRP, más general por trabajar con varias rutas, existen dos posibles opciones: a) restringir la actuación del operador a elementos dentro de una misma ruta, es decir, a los elementos pertenecientes al mismo vehículo, o b) permitir que los operadores trabajen con cualquier elemento independientemente de que esté o no en la misma ruta. Se optó por la segunda para no limitar la capacidad de modificación de los operadores.

Cualquiera de los operadores descritos puede usarse para mutar tanto el vector permutación como el vector de reabastecimientos, sin embargo, en el caso de los reabastecimientos tienen una utilidad bastante limitada, pues al cambiar solo las posiciones de los elementos ya existentes, no permiten eliminar o añadir nuevos repostajes. Para abordar la mutación de esta parte de la solución se propusieron dos operadores: 1) la inversión binaria de un único elemento y 2) la aplicación aleatoria de una acción de añadir, eliminar o cambiar. La primera estrategia consiste en escoger al azar uno de los elementos del vector y aplicarle la inversión binaria, de este modo se garantiza que en cada paso solo se añade o elimina un reabastecimiento. La segunda estrategia escoge al azar una de las siguientes tres posibles acciones: añadir, eliminar o cambiar de lugar un reabastecimiento, y la lleva a cabo. En el primer caso se selecciona al azar cualquiera de las componentes con valor 0 del vector y se cambia por un 1. El segundo caso es similar salvo que con 1's en lugar de 0's. Para el cambio de lugar, se escoge un reabastecimiento al azar y se mueve a otra componente cualquiera que no contenga ya uno. Ambas producen pocos cambios y garantizan que el número de reabastecimientos varíe lentamente. La Figura 3.12 muestra un ejemplo de ambos tipos de mutación.

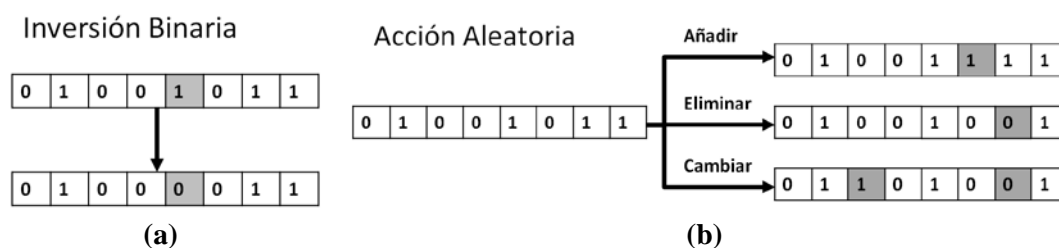


Figura 3.12. Operadores de mutación para el vector de reabastecimientos. (a) Inversión binaria y (b) acción aleatoria.

3.2.5.2.1. Operador Mix

Los operadores de mutación considerados para el vector permutación parecen ser buenas estrategias para construir nuevas soluciones, sin embargo no es inmediato

discernir cuál de ellos es el mejor, ya que cada uno presenta ventajas según el momento en el que se encuentre la búsqueda. No obstante, si se estudian con detenimiento se observa que todos ellos pueden expresarse en términos de las mismas operaciones elementales sencillas (desplazamientos, intercambios, inversiones, etc.) que en muchos casos comparten. Por ejemplo, tanto el operador *relocate* como el operador *or-opt* aplican un desplazamiento, aunque el primero mueve solo un elemento mientras que el segundo varios. Asimismo, el operador *cut inversion* también aplica un desplazamiento de varios elementos, pero invirtiendo el orden de los elementos, inversión que también emplea el operador *simple inversion*. Respecto al operador LSM, también se usan desplazamientos, pero en este caso se prueba como quedarían los elementos a mover en diferentes sitios y se escoge la opción que mejore más la solución. Por último, el operador *swap* aplica un intercambio y un intercambio puede entenderse como dos desplazamientos simultáneos. Partiendo de estas consideraciones, se concibió un nuevo operador que aglutinara todas las operaciones elementales en las que se basan, de tal modo que pudiera comportarse como cualquiera de los operadores de mutación anteriores. En particular, el nuevo operador, llamado *mix-opt*, se diseñó para que agrupara las siguientes operaciones elementales:

- 1) Desplazamiento de un elemento o desplazamiento unitario: selecciona un elemento aleatoriamente y lo desplaza a otra posición también escogida aleatoriamente.
- 2) Intercambio de elementos o intercambio unitario: intercambio de dos elementos elegidos aleatoriamente.
- 3) Desplazamiento de un segmento: desplazamiento de un grupo consecutivo de elementos (segmento) seleccionado de forma aleatoria. Similar al desplazamiento unitario pero con un segmento.
- 4) Intercambio de segmentos: intercambio de dos segmentos escogidos aleatoriamente. Similar al intercambio unitario pero usando segmentos.
- 5) Desplazamiento de segmentos entre vehículos: desplazamiento de un segmento, cuyos elementos pertenecen todos a la ruta de un mismo vehículo, en la ruta de otro vehículo.

En definitiva, la idea consiste en que cada vez que este operador se aplique, se ejecute una cualquiera de estas operaciones individuales y de este modo la mutación de soluciones no se limite a un único tipo de cambio. La probabilidad de cada una de ellas de ser aplicada es $w_i / (\sum_{i=1}^5 w_i)$, siendo w_i el peso asignado a la operación i . La ecuación (3.28) describe matemáticamente al operador definido siendo $r \in [0, \sum_{i=1}^5 w_i]$ un valor generado de forma aleatoria.

Los pesos determinan que operaciones se ejecutan con más frecuencia y deben ser por tanto ajustados en función de la importancia que se le desee dar a cada operación. La idea es permitir que se puedan aplicar diferentes operaciones ya que todas ellas han demostrado ser útiles en la literatura, aunque cada una de ellas sea más idónea para algunas etapas de exploración que para otras. Por ejemplo, el desplazamiento unitario y el intercambio unitario son similares a los operadores *relocate* y *swap*, y son útiles en etapas avanzadas de la búsqueda, cuando es necesario refinar las soluciones y se requieren cambios pequeños y precisos. Por otro lado, las operaciones con segmentos son muy útiles en etapas tempranas, cuando las soluciones requieren muchos y grandes cambios,

de ahí que sea interesante contar con operaciones que usen más elementos. Es interesante observar que la operación desplazamiento de segmentos es similar al operador *or-opt*.

$$\begin{aligned}
 &mix - opt(x) = \\
 &= \left\{ \begin{array}{ll}
 \text{desplazamiento unitario}(x), & \text{if } r \leq \sum_{i=1}^1 w_i \\
 \text{intercambio unitario}(x), & \text{else if } r \leq \sum_{i=1}^2 w_i \\
 \text{desplazamiento de segmento}(x), & \text{else if } r \leq \sum_{i=1}^3 w_i \\
 \text{intercambio de segmentos}(x), & \text{else if } r \leq \sum_{i=1}^4 w_i \\
 \text{desplazamiento de segmento entre vehiculos}(x), & \text{else } r \leq \sum_{i=1}^5 w_i
 \end{array} \right. \quad (3.28)
 \end{aligned}$$

La operación intercambio de segmentos se incluyó para tener una operación capaz de producir más cambios incluso que el operador de desplazamiento de segmentos (un intercambio implica dos desplazamientos). Por último, la operación de inserción de segmentos entre vehículos se añadió porque el problema que se quiere resolver involucra a más de un vehículo y parece apropiado contar con un modo de transferir fácilmente rutas o parte de ellas directamente de un vehículo a otro. Esta operación es especialmente interesante para cuando se optimicen criterios, como el tiempo, que siempre hacen uso de todos los vehículos a la vez, porque agiliza el reparto del trabajo, es decir, la transferencia de pistas entre toda la flota. El considerar vehículos heterogéneos también da sentido a esta operación, porque puede suceder que una ruta que no sea óptima para un determinado vehículo, sí lo sea para otro con distintas características y gracias a esta operación es muy fácil transferir la ruta del primer vehículo al segundo.

Por analogía, también se consideró una operación de intercambio de segmentos entre vehículos en un primer momento, pero finalmente acabó siendo descartada por no ser necesaria, ya que el número de vehículos, en términos generales, suele ser pequeño y no parece necesario tener más operadores que permitan la transferencia de rutas entre vehículos, ya que esto podría limitar la convergencia.

Es interesante observar que los intercambios introducen más cambios que los desplazamientos al igual que las operaciones de segmentos alteran más que los operadores unitarios. Puesto que el nuevo operador combina todas ellas aleatoriamente, es capaz de aplicar cambios de diferentes grados (muy pequeños, pequeños, medianos, grandes, etc.) que se ajustan a las diferentes necesidades que surgen a lo largo de todo el proceso de convergencia.

Uno de los aspectos del nuevo operador a tener en cuenta es el número de elementos de las operaciones con segmentos, es decir, la longitud de los segmentos. La mayoría de los operadores que usan varios elementos no especifican el número exacto que se debe usar y, sin embargo, es un factor que puede afectar a la convergencia, ya que cuanto más largo sea un segmento más grandes son los cambios y, consecuentemente, mejor se comporta el operador en etapas tempranas de búsqueda. Por el contrario, los segmentos más cortos benefician más las etapas finales de refinamiento. En definitiva, la longitud óptima no está clara por lo que es un parámetro que debe ser configurado antes de usar el operador.

Además de la longitud de los segmentos, conviene especificar si es posible o no invertirlos, de este modo se incluyen las características de los operadores *simple inversion* y *cut inversion*.

Por último, para incluir también las propiedades del operador *LSM*, debería ser posible especificar si el operador puede realizar búsquedas locales, como las del *LSM*. Esto es, si cuando escoge una operación, ésta puede intentarse varias veces, generando varias soluciones, y seleccionando la mejor como resultado final.

Si se consideran todas estas opciones de ejecución, el nuevo operador mix-opt puede comportarse como cualquiera de los operadores de la literatura descritos en la sección anterior. Ahora bien, para el buen funcionamiento del operador propuesto resulta imprescindible configurar a priori los aspectos o variables que afectan a la convergencia del mismo que se acaban de describir. La Tabla 3.6 muestra estas variables y sus rangos de valores.

Tabla 3.6. Variables de configuración del nuevo operador

Operación	Longitud (%) $\in [0, 100)$	Nº de intentos $\in [0, \infty)$ $\cup \{\text{todos}\}$	Inversión $\in \{\text{falso}, \text{cierto}\}$	Peso $\in [0, \infty)$
Desplazamiento unitario	l_1 (= 1 elemento)	a_1	r_1 (= falso)	w_1
Intercambio unitario	l_2 (= 1 elemento)	a_2	r_2 (= falso)	w_2
Desplazamient. de segmento	l_3	a_3	r_3	w_3
Intercambio de segmento	l_4	a_4	r_4	w_4
Desplazamient. de segmento entre vehículos	l_5	a_5	r_5	w_5

Donde l_i especifica el tamaño máximo del segmento respecto al número total de elementos, por ejemplo, si l_i es igual a 50 %, indica que el segmento con el que se trabaje tendrá entre 1 y la mitad de elementos del vector de entrada. l_1 y l_2 son siempre igual a 1 elemento porque, por definición, las operaciones unitarias solo manejan uno. El número de nuevas soluciones que se generan cuando se realiza una búsqueda local lo determina a_i . El valor *todos* indica que se probará a situar el elemento o segmento de trabajo en todas las posiciones posibles, por ejemplo, si un vector permutación tiene 10 elementos y se debe aplicar un desplazamiento unitario y el número de intentos es *todos*, el elemento seleccionado se probará en las 9 posiciones posibles restantes y la que produzca la mejor solución será seleccionada. Del mismo modo, si r_i es igual a *cierto*, en cada operación con segmentos, estos se probarán tanto en su orden original como en el inverso, seleccionando aquel que produzca una solución mejor. Asimismo, r_1 y r_2 son siempre *falso* porque no ser necesarios ya que el inverso de un segmento de un único elemento es siempre el mismo segmento. Por último, los w_i determinan la probabilidad de ejecutar cada operación, de modo que cuanto mayor es el peso más veces el mix-opt seleccionará y ejecutará la operación asociada.

Debido a la gran cantidad de parámetros que hay que ajustar, en este trabajo de investigación se ha abordado la búsqueda de una buena configuración general que se pueda usar por defecto (sección 3.3.1). Asimismo, se han probado diferentes variantes del nuevo operador descartando algunas de sus operaciones con la finalidad de evaluar en qué

medida influyen en el rendimiento del operador, es decir, su impacto en la convergencia, así como para comprobar lo dependiente que es el operador de la configuración escogida (sección 3.4.1).

3.2.6. Reemplazando el conjunto de trabajo

Dependiendo de la meta-heurística, la sustitución del conjunto de soluciones (conjunto de trabajo) que se están evaluando por el nuevo conjunto de soluciones a evaluar es diferente. El Recocido Simulado, cuyo conjunto de trabajo está formado por una única solución, siempre selecciona la nueva solución si ésta es mejor que la solución actual, si no lo es, todavía es posible aceptar la nueva solución dependiendo de una función de probabilidad que decrece con el tiempo. Esta estrategia es el mecanismo que utiliza el Recocido Simulado para escapar de los mínimos/máximos locales.

En el Algoritmo Genético se usa habitualmente alguna de las siguientes estrategias de reemplazo: 1) sustituir directamente el conjunto presente de soluciones por el nuevo conjunto, 2) sustituir el conjunto presente por el nuevo, pero sustituyendo la peor solución del nuevo conjunto por la mejor solución del conjunto presente (estrategia conocida como elitismo de un único elemento), o 3) unir el conjunto presente de soluciones y el nuevo conjunto y seleccionar las n mejores soluciones, siendo n el número de soluciones que debe tener el conjunto de trabajo. Esta variante se conoce como elitismo poblacional. Esta última estrategia es la que usa el método NSGA-II junto con el criterio de dominancia de *pareto*, que veremos más adelante (sección 3.2.9), para determinar cuáles son las n mejores soluciones.

3.2.7. Recocido Simulado

El nombre e inspiración de este método, Recocido Simulado (Kirkpatrick, Gelatt, & Vecchi, 1983) (Cerny, 1985), viene del proceso de recocido y enfriamiento de metales en metalurgia, una técnica que consiste en calentar y luego enfriar lentamente un material para moldearlo con facilidad. El calor aporta la energía suficiente para que los átomos se muevan y el material se haga más maleable, así puede organizarse con una nueva estructura interna o estado, mientras que el enfriamiento paulatino y controlado hace que los cambios sean cada vez más pequeños y localizados, y que el material acabe cristalizando en un estado final con una estructura interior, por lo general, más organizada que la inicial.

La analogía con el esquema típico de optimización iterativa de las meta-heurísticas consiste en que cada estado representa una solución, su energía asociada (que depende de la estructura interna de sus átomos) es el valor objetivo de la solución (que análogamente depende de sus variables de decisión internas), y las transiciones entre estados son los saltos que los operadores producen para recorrer el espacio de búsqueda. El objetivo, por tanto, es encontrar el estado que tenga la estructura interna con menor energía asociada, es decir, la solución que tenga el menor o mayor valor objetivo, dependiendo del tipo de proceso de optimización.

Extrapolando a una búsqueda computacional, el algoritmo parte de un estado inicial s con una energía inicial $E(s)$ y está constantemente generando un nuevo estado s' con cierta aleatoriedad mediante el operador de construcción de estados vecinos. La función de aceptación es quien determina si la búsqueda transita al nuevo estado s'

evaluando la diferencia de energía entre el estado actual y el nuevo estado, $\delta E = E(s') - E(s)$. Si el valor es negativo y suponemos que estamos en una minimización, se acepta el tránsito, es decir, siempre se acepta el salto a un estado mejor. En el caso de que el valor de δE sea positivo, también es posible aceptar nuevos estados, aun siendo peores, dependiendo de una probabilidad $P(\delta E, T) = \exp(-\frac{\delta E}{T})$ que aumenta cuanto menor es la diferencia δE y mayor la temperatura T , tal como se muestra en la Figura 3.13. En la gráfica se observa que cuando el algoritmo se enfría totalmente ($T = 0$), la probabilidad de transición a un estado peor tiende asintóticamente a cero y solo se aceptan estados mejores, de tal modo que la búsqueda se vuelve similar a la de un algoritmo voraz, mientras que para valores δE muy pequeños, la probabilidad de aceptación es muy alta.

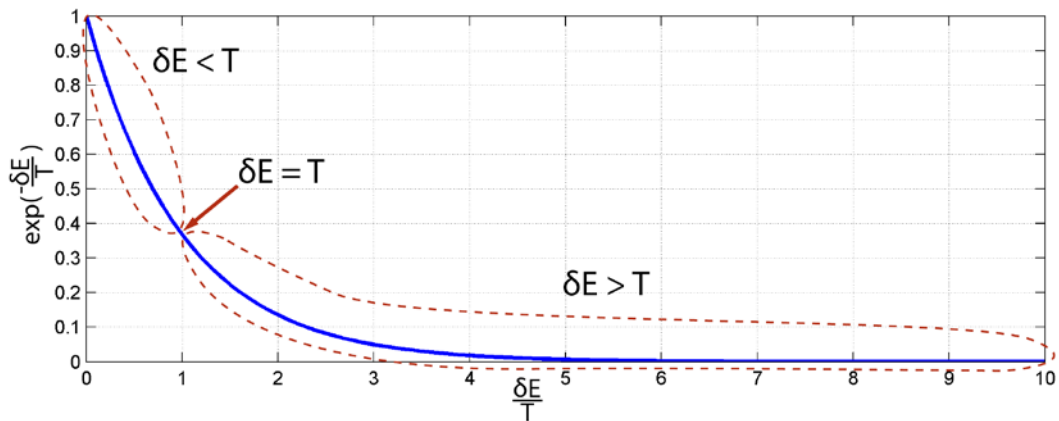


Figura 3.13. Rango de la función $\exp(-\frac{\delta E}{T})$ cuando δE es positiva (por tanto, cambios a peor) y $\frac{\delta E}{T}$ se mueve entre 0 y 10.

De todo esto se desprende que la variable temperatura T juega un papel muy importante en el control de la evolución de la búsqueda. A temperaturas altas, los saltos de exploración pueden ser grandes y en cualquier dirección, mientras que a temperaturas más bajas, los cambios son menores y se explora de un modo más local.

Volviendo a la terminología general, aunque en un primer momento pueda parecer contraproducente aceptar soluciones que no solo no mejoran los resultados de la búsqueda, sino que los empeoran, si se usa convenientemente, este comportamiento puede ayudar a la convergencia permitiendo escapar de mínimos/máximos locales cuya única salida pasa por transitar temporalmente por soluciones peores. No obstante, como las soluciones mejores son siempre aceptadas y la temperatura decrece con el tiempo hasta no permitir cambios a peor, el algoritmo garantiza que, convenientemente configurado, tarde o temprano convergerá hacia soluciones mejores y, tras cierto tiempo que depende del tamaño del espacio de búsqueda, las soluciones encontradas estarán claramente cerca del óptimo. En definitiva, dada la importancia de la configuración en posteriores secciones (3.3.2) se explica cómo se ha abordado el ajuste de este método.

Sobre la disminución de la temperatura T , la forma más habitual de implementar este comportamiento es multiplicar el parámetro por un factor de enfriamiento $\alpha < 1$ en cada iteración. El ajuste adecuado de este factor es importante porque cuanto más lento es

el enfriamiento mayor es el número de iteraciones en las que se aceptan soluciones peores y, por tanto, puede ralentizar la convergencia.

```

1  FUN  $s_{opt}$  = recocidoSimuladoConRecalentamiento( $s_0, T_0, t_e, \alpha$ )
2       $s = s_0$ ;
3       $s_{opt} = s$ ;
4       $T = T_0$ ;
5       $t_{aceptacion} = tiempoActual()$ ;
6      WHILE (NOT condicionDeTerminacion( $s, \dots$ ))
7           $s' = estadoVecino(s)$ ;
8          IF (aceptar( $E(s), E(s'), T$ )) THEN
9              IF ( $E(s') < E(s)$ ) THEN
10                  $t_{aceptacion} = tiempoActual()$ ;
11             END
12              $s = s'$ ;
13             IF ( $E(s) < E(s_{opt})$ ) THEN
14                  $s_{opt} = s$ ;
15             END
16         END
17         IF (recalentar( $t_{aceptacion}, t_e$ )) THEN
18              $T = T_0$ ;
19         ELSE
20              $T = enfriar(T, \alpha)$ ;
21         END
22     END
23     RETURN  $s_{opt}$ ;
24 END
25
26 FUN bool aceptar( $e, e', T$ )
27     RETURN ( $e' < e$  OR  $\exp(-(e'-e)/T) > valorAleatorio()$ );
28 END
29
30 FUN bool recalentar( $t_{aceptacion}, t_e$ )
31     RETURN ( $tiempoActual() - t_{aceptacion} > t_e$ );
32 END
33
34 FUN float enfriar( $T, \alpha$ )
35     RETURN  $T \cdot \alpha$ ;
36 END

```

Figura 3.14. Recocido Simulado con recalentamiento

Por último, para evitar estancamientos en mínimos locales imposibles de abandonar cuando la temperatura T es baja, existe la posibilidad de *recalentar* siguiendo la estrategia propuesta por (Ingber, 1989). Recalentar consiste en subir la temperatura T cuando el algoritmo lleva bastante tiempo estancado probablemente en un mínimo/máximo local, de este modo es posible reanudar la búsqueda en cualquier dirección, incluso aunque la temperatura se haya enfriado totalmente, y escapar más tarde o más temprano de aquellos mínimos/máximos locales en los que se haya cristalizado. El tiempo que se puede estar estancado antes de recalentar es otra variable importante

porque determina el tiempo que se está explorando una determinada zona del espacio antes de saltar a otra. Valores pequeños pueden producir exploraciones excesivamente aleatorias, es decir, con continuos saltos a soluciones peores, sin tiempo suficiente para explorar apropiadamente cada región, mientras que valores muy grandes pueden generar búsquedas demasiado localizadas y exhaustivas, ralentizando la convergencia.

Al igual que sucede con la temperatura, el factor de enfriamiento y el tiempo de estancamiento también deben configurarse convenientemente como se analizará más adelante (sección 3.3.2).

El listado de pseudo-código de la Figura 3.14 muestra como se ha adaptado en este trabajo de investigación el funcionamiento del Recocido Simulado para incluir la opción de recalentamiento. En este listado de código, s_0 es el estado inicial, T_0 la temperatura inicial, t_e el tiempo máximo que se puede estar estancado antes de recalentar, α el factor de enfriamiento de la temperatura T , s_{opt} el estado mejor que encuentra el algoritmo, $t_{acceptacion}$ el tiempo en el que se aceptó el último salto de exploración hacia una solución mejor, *condicionDeTerminacion(s,...)* indica si se cumple alguno de los criterios de parada del algoritmo (solución suficientemente buena, límite de estancamiento o máximo intervalo de tiempo sin mejorar la solución superado, etc.), *estadoVecino(s)* es el operador que construye el estado vecino s' (nueva solución) a partir del estado actual s (solución actual), *tiempoActual()* devuelve el tiempo exacto, con precisión de milisegundos, en el mismo instante de la llamada y *valorAleatorio()* devuelve un valor aleatorio entre 0 y 1.

3.2.8. Algoritmos Genéticos

El Recocido Simulado explora el espacio de búsqueda usando una única solución candidata que va saltando de región en región. A este tipo de meta-heurísticas se las llama de punto único o basadas en trayectoria, porque la solución candidata va dibujando una trayectoria por el espacio de búsqueda. Existen otras estrategias de búsqueda que son capaces de explorar en cada iteración más de una solución, son las conocidas como basadas en población. Estas meta-heurísticas tienen a su favor, respecto a las de un único punto, que trabajan con más de un candidato en cada paso, lo que, a priori, les otorga más capacidad de rápida convergencia por disponer de más candidatos de los que extraer buenas propiedades.

Entre las estrategias de exploración del espacio de búsqueda basadas en población más conocidas, se encuentran los algoritmos genéticos (Goldberg & Holland, 1988), que se inspiran en el proceso de selección natural y en su base genética. En el libro *El origen de las especies* (Darwin, 1859), Charles Darwin resume su teoría de la evolución por selección natural de la siguiente manera: “Si existen organismos que se reproducen y la progenie hereda características de sus progenitores, y existen variaciones de características, y el medio ambiente no admite a todos los miembros de una población en crecimiento, entonces aquellos miembros de la población con características menos adaptadas (según lo determine su medio) morirán con mayor probabilidad mientras que aquellos miembros con características mejor adaptadas tendrán mayor probabilidad de supervivencia”.

En definitiva, los Algoritmos Genéticos implementan un proceso de optimización iterativo en el que se parte de una población inicial de individuos que se hace evolucionar por medio de operaciones de cruce y mutación (que emulan los procesos de reproducción y mutación habituales en la naturaleza) para dar lugar a nuevas poblaciones con

individuos más adaptados. Más detalladamente y en términos algorítmicos, en cada iteración se genera una nueva población P y se calcula el vector F_P que contiene el nivel de aptitud de cada uno de los individuos, los llamados valores *fitness*, mediante la función de aptitud, o función de fitness, que indica lo bueno que es un individuo en términos del criterio de optimización (en el proceso real, es la medida de cómo de adaptado está el individuo al medio). Posteriormente estos valores en F_P se utilizan para seleccionar los individuos progenitores Q a partir de los cuales se genera la nueva población P' . Generalmente la probabilidad de seleccionar un individuo suele ser directamente proporcional a su valor de aptitud, de tal modo que los mejores individuos tienen mayor probabilidad de generar descendencia, como suele ocurrir en la naturaleza. Una de las técnicas de selección de progenitores más conocida es la conocida como la ruleta, por estar inspirada en la ruleta de un casino. Esta técnica simula una ruleta virtual donde a cada individuo se le asigna una porción con un tamaño proporcional a su valor de aptitud, por lo que cuanto mejor sea el valor mayor es la porción asignada y, por tanto, mayor es la probabilidad de que el individuo sea seleccionado.

Una vez obtenida la población de progenitores Q , se genera la nueva población P' mediante cruces entre los padres hasta construir una población de igual tamaño a la inicial y en la que, además, parte de los individuos sufren mutaciones. La cantidad de individuos que provienen de cruces y mutaciones se establece a partir de las probabilidades de cruce p_c y de mutación p_m . Probabilidades de cruce altas favorecen poblaciones más endogámicas mientras que probabilidades de mutación altas favorecen la diversidad.

Una vez generada, la nueva población también se evalúa y se obtienen los valores de aptitud $F_{P'}$ para cada individuo, y se reemplaza la población actual P por alguna combinación entre la propia población P y la nueva población P' . Un mecanismo adicional que se utiliza con cierta frecuencia en el reemplazo, como ya se comentó anteriormente, es el elitismo o posibilidad de copiar directamente los mejores individuos de la población progenitora en la nueva población, lo que garantiza que en la nueva población haya como mínimo un individuo tan bueno como el mejor de la población progenitora, situación que favorece la convergencia.

Respecto a la terminación del algoritmo, al igual que con el Recocido Simulado, el proceso se itera una y otra vez hasta que se cumple algún criterio de parada previamente definido, donde los más habituales son: exceder un número determinado de tiempo o iteraciones, sobrepasar un determinado valor umbral que se considera suficientemente cercano al óptimo y permanecer un determinado tiempo o número de iteraciones sin que el valor objetivo de la mejor solución de trabajo mejore (estancamiento).

El listado de pseudo-código de la Figura 3.15 muestra el Algoritmo Genético descrito. La variable n representa el tamaño de población, s el tipo de técnica a aplicar para la selección de progenitores, p_c la probabilidad de cruce, p_m la probabilidad de mutación y e el tipo de elitismo. Todas estas variables deben ajustarse previamente a la ejecución y pueden ralentizar o acelerar la convergencia, por ello la tarea de ajuste es importante y se aborda más adelante (sección 3.3.3). La función *poblacionInicial*(n) crea una población inicial aleatoria de n individuos, *evaluar*(P) evalúa la población P y devuelve un vector F_P con los valores de aptitud de todos los individuos, *seleccionarPadres*(s, P, F_P) escoge un conjunto de progenitores de entre los individuos de la población P dando mayor probabilidad a aquellos con mejores valores de aptitud, *seleccionNatural*($e, P, F_P, P', F_{P'}$) filtra los mejores individuos de entre las poblaciones P y P' usando un elitismo e , *condicionDeTerminacion*(s, \dots) indica si se cumple alguno de los

criterios de parada del algoritmo (solución suficientemente buena, tiempo de estancamiento superado, etc.) y *valorAleatorio()* devuelve un valor aleatorio entre 0 y 1.

```

1  FUN P = algoritmoGenetico(n,s,pc,pm,e)
2    P = poblacionInicial(n);
3    WHILE (NOT condicionDeTerminacion(P,...))
4      Fp = evaluar(P);
5      Q = seleccionarPadres(s,P,Fp);
6      P' = cruzar(Q,pc);
7      P' = mutar(P',pm);
8      Fp' = evaluar(P');
9      P = seleccionNatural(e,P,Fp,P',Fp');
10   END
11   RETURN P;
12 END
13
14 FUN P = cruzar(Q, pc)
15   P = {};
16   FOR (padrei, padrei+1) IN Q
17     IF (valorAleatorio() <= pc) THEN
18       H = cruce(padrei, padrei+1);
19       P = P U H;
20     ELSE
21       P = P U {padrei, padrei+1};
22     END
23   END
24   RETURN P;
25 END
26
27 FUN P = mutar(Q,pm)
28   P = {};
29   FOR q IN Q
30     IF (valorAleatorio() <= pm) THEN
31       P = P U {mutacion(q)};
32     ELSE
33       P = P U {q};
34     END
35   END
36   RETURN P;
37 END

```

Figura 3.15. Algoritmo Genético

3.2.9. Non-Dominated Sorting Genetic Algorithm II

Tanto el Recocido Simulado como el Algoritmo Genético solo se pueden utilizar para encontrar soluciones que optimicen un único criterio. Ahora bien, como uno de los objetivos de este trabajo es optimizar la tarea agrícola para varios costes a la vez, se hace imprescindible trabajar con alguna técnica multi-objetivo. Sin salir de los algoritmos evolutivos, existe una meta-heurística llamada *non-dominated sorting genetic algorithm*

II (Deb, Agrawal, Pratap, & Meyarivan, 2000), también conocida como NSGA-II capaz de tratar con varios objetivos simultáneamente. La estructura de funcionamiento es muy similar a la de los Algoritmos Genéticos, pero en lugar de evaluar la aptitud de cada individuo para un solo objetivo se evalúa la aptitud para varios objetivos. En consecuencia, cada individuo tiene diferentes valores de aptitud cada uno asociado a un criterio, y no es tan fácil establecer una relación de orden entre los individuos, lo que obliga a adaptar la etapa de selección. En otras palabras, es necesario usar algún tipo de mecanismo capaz de discernir qué individuos son mejores considerando todos sus valores de aptitud en conjunto. El método NSGA-II resuelve esta cuestión mediante el criterio de *dominancia de Pareto* (Pardalos, Migdalas, & Pitsoulis, 2008) que establece el criterio de dominancia enunciado en (3.29).

Dominancia de Pareto: Dados dos vectores de m valores objetivo $\mathbf{u} = (u_1, \dots, u_m)$ y $\mathbf{v} = (v_1, \dots, v_m)$, se dice que \mathbf{u} domina \mathbf{v} si y sólo si:

$$\forall i \in \{1, \dots, m\}, u_i \leq v_i \wedge \exists j \in \{1, \dots, m\} \mid u_j < v_j \quad (3.29)$$

En otras palabras, un individuo domina a otro si y solo si todos sus valores de aptitud son mejores o iguales que los del otro y al menos uno de ellos es estrictamente mejor. Asimismo, un individuo es un óptimo de Pareto si y solo si no existe otro individuo en el espacio de búsqueda que lo domine.

En general la solución óptima en términos de Pareto a un problema de multi-optimización no suele ser única, si no que está formada por el conjunto de todas las soluciones no dominadas del espacio de búsqueda. A este conjunto se le llama *frente de Pareto*. La Figura 3.16 muestra los diferentes frentes de Pareto (líneas azul oscuro) según el objetivo sea maximizar o minimizar las funciones f_1 y f_2 , siendo la imagen de ambas funciones la superficie delimitada por la circunferencia (área azul).

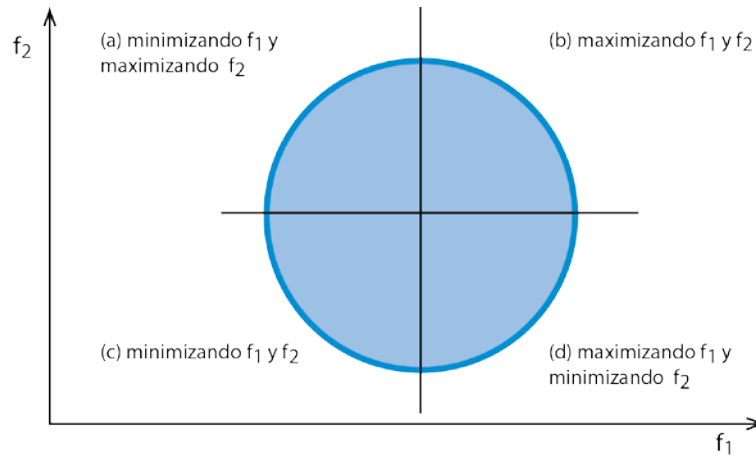


Figura 3.16. Frentes de Pareto (líneas azul oscuro) para las funciones f_1 y f_2 que producen la imagen acotada por una circunferencia, cuando los objetivos son (a) minimizar f_1 y maximizar f_2 , (b) maximizar f_1 y maximizar f_2 , (c) minimizar f_1 y minimizar f_2 , y (d) maximizar f_1 y minimizar f_2 .

Así gracias al criterio de dominancia, se puede establecer una primera relación de orden entre un grupo de individuos dividiendo el grupo en frentes y ordenándolos según el frente al que pertenecen. El primer frente estaría formado por los individuos no dominados, el segundo por los no dominados entre los individuos restantes (grupo original menos el primer frente), y así sucesivamente hasta no quedar ningún individuo sin ordenar. Ahora bien, como los frentes pueden contener más de un individuo, es necesario establecer una segunda relación de orden para los individuos de un mismo frente. Para estos casos, el método NSGA-II usa la distancia de *crowding* (apiñamiento) que es un indicador de lo cerca que está un individuo de sus vecinos. Dado un individuo, la distancia de crowding, es la suma de todas las distancias normalizadas con los vecinos inmediatamente superiores o inferiores, y da una idea de lo aislado que está el individuo dentro del frente. El NSGA-II interpreta ese aislamiento como una estimación de lo diferente que es ese individuo respecto a los demás y, a igualdad de frentes, siempre escoge a los individuos con mayor distancia de *crowding* para favorecer la diversidad de los individuos de la población.

La Figura 3.17 muestra los diferentes frentes para un grupo de individuos dispuestos según dos valores objetivos y la distancia de *crowding* de uno de ellos.

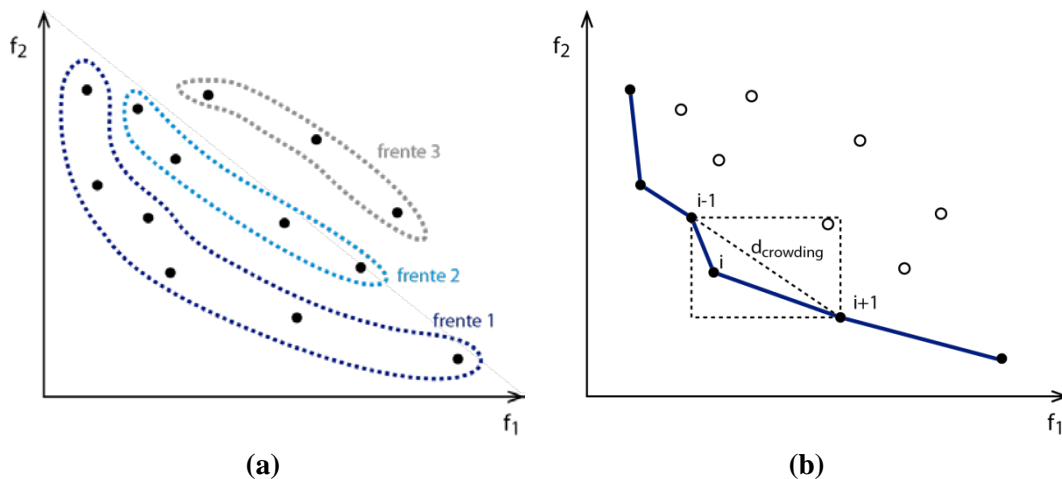


Figura 3.17. (a) Frentes para un grupo de individuos dispuestos según dos valores objetivos que se deben minimizar, y (b) distancia de crowding del individuo i del primer frente.

Gracias a las relaciones anteriores, el método NSGA-II es capaz de establecer un orden entre las soluciones de la población que permite implementar la etapa de selección. El mecanismo de selección es la mayor diferencia respecto a los Algoritmos Genéticos.

Otras diferencias son que en el método NSGA-II la selección de los progenitores siempre se realiza mediante un torneo (Miller & Goldberg, 1995), que consiste en elegir varios individuos al azar, habitualmente dos, y quedarse con el mejor, es decir, con el que tiene el menor frente y la mayor distancia de *crowding* a igualdad de frente. El proceso de selección se repite hasta que se tienen suficientes progenitores para generar una nueva población progenitora que, a diferencia de los Algoritmos Genéticos, no tiene porque ser del mismo tamaño que la población de partida.

Al igual que en un Algoritmo Genético, la nueva población se crea mediante cruces y mutaciones. En este caso no existen distintos tipos de reemplazo sino que se une siempre a la población actual y se aplica el elitismo poblacional de los Algoritmos Genéticos, es decir, se unen la población actual y la nueva población y se ordenan todos los individuos por frente y por distancia de *crowding*, pasando a la siguiente iteración los n individuos mejores, siendo n el tamaño de la población de trabajo.

Del mismo modo que en los anteriores métodos, el proceso global se repite hasta que se cumple alguno de los criterios de terminación.

El listado de pseudo-código de la Figura 3.18 muestra el algoritmo, muy similar al de un Algoritmo Genético (véase pseudo-código de la Figura 3.15). El tamaño de la población actual es n , el tamaño de la nueva población es n_h , la probabilidad de cruce es p_c y la probabilidad de mutación es p_m . Al igual que en los métodos anteriores, el valor de estas variables afecta a la convergencia de la búsqueda por lo que es esencial elegir los valores adecuadamente tal como se muestra más adelante (Sección 3.3).

```

1  FUN P = NSGA-II( $n, n_h, p_c, p_m$ )
2      P = poblacionInicial( $n$ );
3       $F_p$  = evaluar(P);
4       $\langle P, F_p \rangle$  = ordenarPorFrenteYCrowding(P,  $F_p$ );
5      WHILE (NOT condicionDeTerminacion(P, ...))
6          Q = seleccionarPadres(P,  $n_h$ );
7          P' = cruzar(Q,  $p_c$ );
8          P' = mutar(P',  $p_m$ );
9           $F_{p'}$  = evaluar(P');
10         R = P  $\cup$  P';
11          $F_R$  =  $F_p \cup F_{p'}$ ;
12          $\langle R, F_R \rangle$  = ordenarPorFrenteYCrowding(R,  $F_R$ );
12         P = R(1: $n$ );
14          $F_p$  =  $F_R$ (1: $n$ );
15     END
16     RETURN P;
17 END

```

Figura 3.18. NSGA-II

La función *poblacionInicial(n)* crea una población inicial aleatoria de n individuos, *evaluar(P)* evalúa la población P y devuelve un vector F_p con los valores de aptitud de todos los individuos, *ordenarPorFrenteYCrowding(P, F_p)* ordena la población P primero por frente y luego por crowding, *seleccionarPadres(P, n_h)* escoge un conjunto con n_h progenitores de entre los individuos de la población P dando mayor probabilidad a aquellos con mejores valores de aptitud, *cruzar(Q, p_c)* y *mutar(P', p_m)* son similares a las funciones ya utilizadas en los Algoritmos Genéticos, y *condicionDeTerminacion(P, \dots)* indica si algún individuo en P cumple alguno de los criterios de parada del algoritmo (solución suficientemente buena, máximo tiempo de búsqueda, etc.).

3.2.10. Comparativa de los métodos

La Tabla 3.7 resume, a modo de comparativa, las principales características de cada uno de los métodos analizados.

Tabla 3.7. Comparativa Recocido Simulado, Algoritmo Genético y NSGA-II

Aspectos principales	Recocido Simulado	Algoritmo Genético	NSGA-II
Tipo de meta-heurística	Basada en trayectoria	Basada en población	Basada en población con multi-objetivo
Nº Soluciones en el conjunto de trabajo	1	n	n
Selección de progenitores	Inmediata	Ruleta, torneo, ranking, etc.	Torneo
Construcción de soluciones	Operador de vecindad	Cruce + mutación	Cruce + mutación
Actualización del conjunto de soluciones	Si mejor, siempre. Si no, dependiente de $P(\delta E, T)$	Copia directa, elitismo individual o elitismo poblacional	Se unen los conjuntos y se seleccionan las n mejores
Nº de objetivos	1	1	m
Relación de orden	<	<	Dominancia de Pareto (frentes y <i>crowding</i>)
Tamaño del conjunto de soluciones de salida	1	n	n (diversidad entre objetivos)
Carga computacional	Baja (respecto a Algoritmo Genético y NSGA-II)	Menor que NSGA-II	Mayor carga computacional que el Recocido Simulado y el Algoritmo Genético

3.3. Obtención de las mejores configuraciones iniciales para los parámetros

Como hemos visto hasta el momento son múltiples las variables que influyen directamente en el proceso de exploración que realiza cada uno de los métodos considerados, parece por tanto recomendable configurar cada método para que trabaje en las mejores condiciones, en otras palabras, usando aquellos valores que más aceleren la convergencia al óptimo. Las estrategias de configuración que se encuentran habitualmente en la literatura consisten en usar valores ya empleados con buenos resultados en otros trabajos o en hacer un ajuste mediante un método de prueba y error. Ambas estrategias se han empleado en el trabajo de investigación que se describe en esta memoria. En concreto, los procesos de configuración de cualquiera de los tres métodos se dividieron en dos etapas consecutivas: 1) un estudio estadístico en el que se evaluaron multitud de configuraciones posibles para tener una primera idea de qué valores permitían alcanzar buenos resultados y cuáles no y 2) una selección de los mejores valores que además encajaban con las recomendaciones existentes en la literatura.

Para el primer paso se muestreó el dominio de las principales variables de configuración de cada método, se formaron todas las configuraciones posibles a partir de las muestras tomadas y se evaluaron repetidamente durante un breve periodo de tiempo sobre un problema sencillo. Al no evaluar todas las configuraciones posibles, sino solo las que se pueden extraer del muestreo y, además, realizar el estudio resolviendo un problema sencillo, no es posible garantizar la obtención de las mejores configuraciones posibles, aún así el proceso seguido tiene interés ya que aporta una orientación empíricamente justificada sobre qué conjuntos de valores pueden producir buenos resultados o cuales tienen claramente un impacto negativo y por tanto conviene evitar en la medida de lo posible.

Además de para los tres algoritmos de optimización, también se llevó a cabo un proceso de configuración sobre el operador propuesto, el mix-opt. Como ya se comentó, este operador requiere la configuración de un gran número de variables antes de ser utilizado en los tres métodos como mecanismo de generación de nuevas soluciones (en el caso del Recocido Simulado, como el operador de vecindad, y en el del Algoritmo Genético y el NSGA-II, como operador de mutación).

El problema utilizado para la configuración consiste en el tratamiento localizado de malas hierbas en un cultivo mediante la pulverización de herbicida. Para ello, se supone que los vehículos de la flota disponen de un mapa con la distribución de las malas hierbas y recorren el campo aplicando herbicida en las zonas indicadas en el mapa. El mapa está dividido en celdas marcadas como libres o como infestadas, y estas últimas tienen que recibir una cantidad de herbicida llamada dosis. Para ello, los vehículos cuentan con un tanque con una cantidad limitada de herbicida (o número limitado de dosis) que, en caso de agotarse, puede reabastecerse acudiendo al depósito.

En concreto, la flota que se empleó en los procesos de configuración estaba formada por 5 vehículos homogéneos con tanques limitados para que ninguno pudiera completar toda la tarea por sí solo y de esta forma asegurar que las soluciones involucraban a más de un vehículo o indicaban la necesidad de reabastecimiento, de este modo el problema era del tipo CVRP y no un mTSP.

Los valores muestreados para cada método y el operador se agruparon en configuraciones, y cada una de ellas se evaluó durante un breve periodo de tiempo, 1 minuto. Asimismo, cada una de estas pruebas se repitió 10 veces extrayendo valores medios como resultado.

Debido a lo breve de cada prueba, el campo considerado se diseñó para contener solo 40 pistas para que la búsqueda tuviese opción a pasar por todas las etapas de convergencia. Normalmente en cualquier búsqueda por aproximación iterativa hay una primera etapa de gran cambio y continua mejora, debido a que las soluciones iniciales se suelen generar aleatoriamente y por tanto son fácilmente mejorables, y una segunda etapa con una convergencia más lenta, por estar próximo a un óptimo, donde ya es más difícil mejorar y con cambios más locales. Empíricamente se comprobó que con 40 pistas y 1 minuto era suficiente para pasar por ambas etapas y que un tiempo inferior o un campo con más pistas daba lugar a un proceso continuo de mejora en el que todas las configuraciones parecían buenas, ya que no se alcanzaba la etapa segunda de convergencia más lenta.

El criterio de evaluación durante los procesos de configuración fue la distancia en cabecera, excepto para el método NSGA-II, donde, se escogieron el coste por insumos y el tiempo requerido para realizar la tarea, por ser el método multi-objetivo. En los primeros casos se seleccionó la distancia por ser un objetivo con mucho impacto en los

demás criterios, puesto que en general minimizar la distancia reduce al menos el coste por combustible y el tiempo requerido para realizar la tarea. Además se particularizó en la distancia en cabecera, porque por construcción las soluciones siempre implican la misma distancia dentro del cultivo, ya que se tiene que recorrer cada pista exactamente una vez, y por tanto, son los recorridos en las cabeceras los que marcan el margen de mejora.

En el caso del NSGA-II se desestimó la distancia, ya que era preferible usar dos criterios claramente contrapuestos como el coste por insumos y el tiempo; obsérvese que el coste por insumos aumenta con cada vehículo extra (por los arranques extra) mientras que el tiempo disminuye.

Para que el problema fuera lo más parecido a una optimización real a pesar de no contar con un número elevado de pistas, se ajustaron algunas de sus características, por ejemplo se utilizó en todos los casos un radio de giro superior al ancho de pista, para que los vehículos tuvieran que saltar como mínimo de tres en tres las pistas para minimizar la distancia dejando pistas entre medias sin tratar. Téngase en cuenta que por la ecuación (2.3), las transiciones entre pistas más juntas requieren maniobras Ω y producen trayectorias más largas. Además, para el radio de giro considerado si solo se usa un vehículo este tiene que dar la vuelta una vez alcanzado el extremo del campo para tratar las pistas restantes, la alternativa es usar más vehículos que vayan cubriendo las pistas dejadas por el primero vehículo. En general, la alternativa reduce la distancia recorrida, tal como se puede ver comparando las Figura 3.19a y Figura 3.19b porque evita los giros más largos que suelen tener que hacer los vehículos en los extremos del campo. Sin embargo, cuando se fuerza a los vehículos a salir de un punto (depósito) y a volver a él a la finalización de la tarea, no está clara la reducción en la distancia recorrida y esta depende de la posición del punto, ya que en este caso el ahorro en giros por usar más de un vehículo se puede ver superado por los recorridos necesarios para salir del punto y volver a él, tal como se ilustra en las Figura 3.19c y Figura 3.19d. Estas dos tendencias enfrentadas hacen que haya dos tipos de soluciones satisfactorias alejadas entre sí en el espacio de búsqueda, lo que implica al menos un mínimo local.

La Figura 3.20 muestra el campo empleado para los procesos de configuración y su distribución de malas hierbas. El campo simula un cultivo de maíz de 40 calles con una separación entre líneas de 0,75 m, con malas hierbas en el interior de las calles cuya distribución viene establecida en un mapa que segmenta el campo en 800 celdas de 0,75 x 0,75 m². De las 800 celdas, 240 son las que hay que tratar por estar infestadas, aproximadamente un 30 %. Se suponen vehículos pequeños con un pulverizador cuyo ancho de trabajo coincide con la calle (por lo que pistas y calles coinciden en este caso), con un tanque de 100 dosis y con un radio de giro de 1 m. Además, el depósito se situó en el punto (15, 18).

Es importante destacar que el problema no hubiera sido más complejo si se hubieran considerado vehículos más grandes, con un ancho de trabajo (y por tanto de pista) mayor, y/o un campo más extenso pero con el mismo número de pistas, puesto que el problema se reduce a recorrer el mismo número de pistas con el mismo número de posibles reabastecimientos intercalados, en otras palabras el espacio de búsqueda sería el mismo ya que es el número de pistas el que determina su tamaño.

La mínima distancia recorrida en cabecera para cubrir el cultivo con las condiciones establecidas se estimó mediante la ecuación (3.24) en 124,75 m y la máxima distancia en 1.533,00 m mediante la ecuación (3.27). A su vez los costes por insumos mínimo y máximo se estimaron en 2,12 € y 4,27 €, respectivamente, y el tiempo mínimo y máximo en 80,98 s y 5.591,88 s, respectivamente.

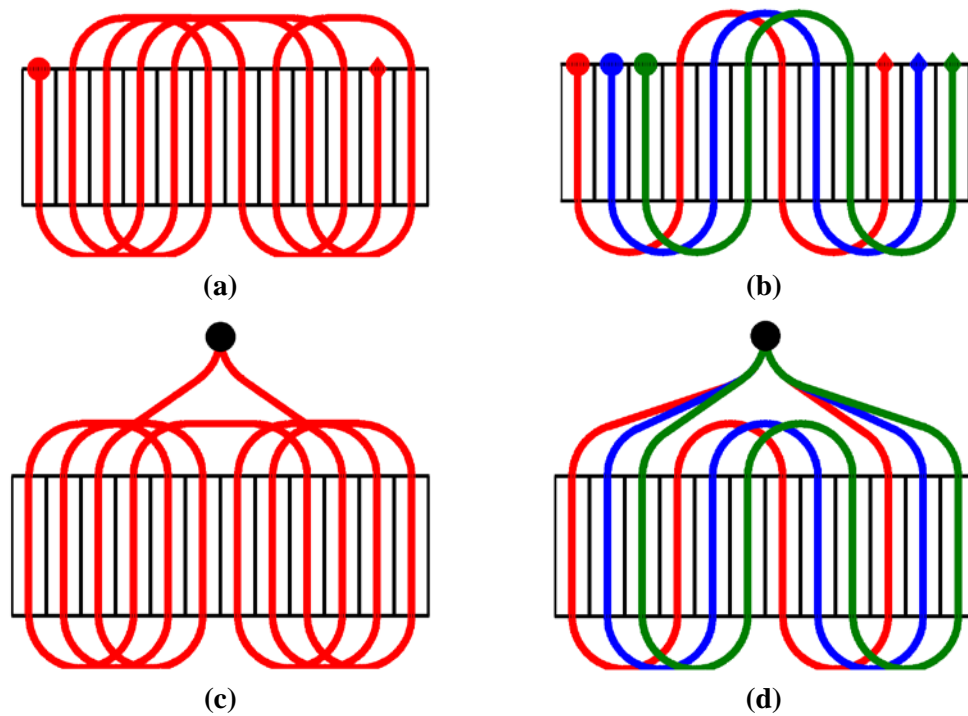


Figura 3.19. El mismo campo cubierto en cuatro casos: con salida y llegada libre por (a) punto de salida y llegada sin fijar y un sólo vehículo, (b) punto de salida y llegada sin fijar y varios vehículos, (c) punto de salida y llegada fijo y un sólo vehículo y (d) punto de salida y llegada fijo y varios vehículos

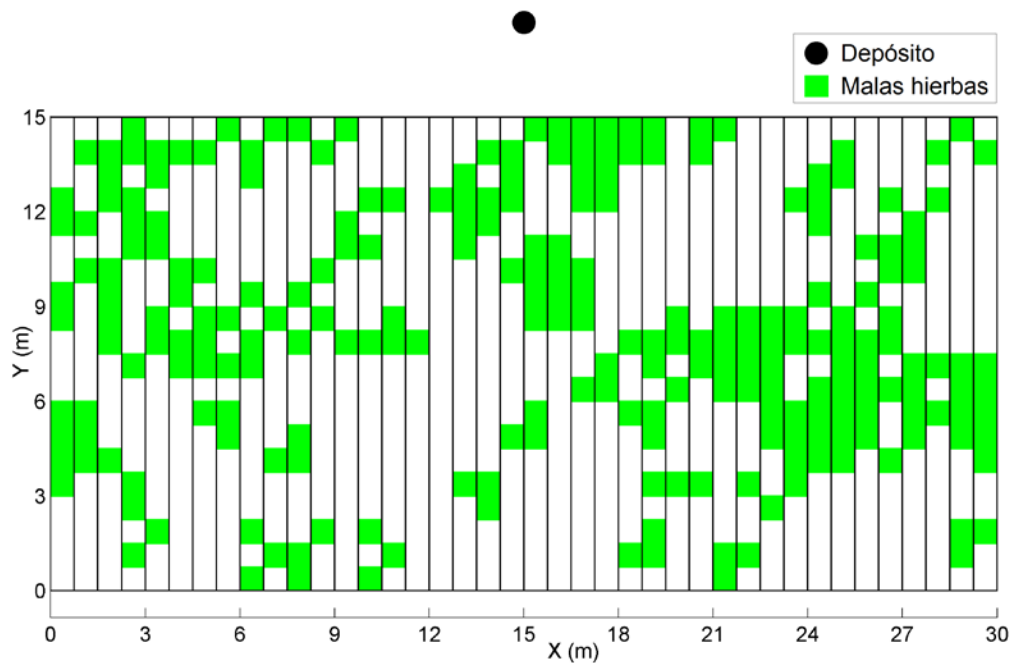


Figura 3.20. Campo utilizado en los procesos de configuración

3.3.1. Configuración del operador mix-opt

En la Tabla 3.8 se muestran las variables de configuración del operador mix-opt junto a los valores que se probaron. Las variables son las longitudes de los segmentos (número de elementos) en las operaciones en las que se usan (l_3, l_4 y l_5), el número de intentos de la búsqueda local asociada a cada operación (a_1, a_2, a_3, a_4 y a_5), la posibilidad de incluir también los segmentos inversos en las búsquedas locales (r_3, r_4 y r_5) y, por último, los pesos para cada operación (w_1, w_2, w_3, w_4, w_5). Debido al gran número de variables, solo se consideraron diferentes valores por operación en de los pesos w_i . Esto significa que la longitud de los segmentos, el número de intentos y la opción de inversión tuvieron que probarse usando el mismo valor todas las operaciones a la vez, ya que de otro modo se habría disparado el número de configuraciones dando como resultado un tiempo de computación inabordable para completar el estudio.

Tabla 3.8. Valores de configuración examinados para las variables del operador mix-opt

Variables	Valores probados
$l(= l_3 = l_4 = l_5) \in L$	$L = \{10\%, 25\%, 50\%, 100\%\}$
$a(= a_1 = a_2 = a_3 = a_4 = a_5) \in A$	$A = \{1, 2, 5, 10, todos\}$
$r(= r_3 = r_4 = r_5) \in R$	$R = \{falso, cierto\}$
$w_1, w_2, w_3, w_4, w_5 \in W$	$W = \{1, 5, 10\}$

Aun con esta reducción se probaron un total de 9.720 configuraciones, que corresponden a $|L| \cdot |A| \cdot |R| \cdot |W|^5 = 4 \cdot 5 \cdot 2 \cdot 3^5$, todas ellas se ejecutaron en paralelo en 10 ordenadores. Como cada prueba duró 1 minuto y cada configuración se ejecutó 10 veces, hicieron falta 6,75 días ($= 9.720 \text{ ejecuciones} \cdot 60 \text{ s}$) de trabajo de cada ordenador para realizar el estudio completo. Si se hubieran probado valores diferentes para cada operación, el número total de configuraciones con el muestreo de la Tabla 3.8 habría sido de $|L|^3 \cdot |A|^5 \cdot |R|^3 \cdot |W|^5 \cong 4 \cdot 10^8$, lo que habría requerido 739,73 años de trabajo de cada ordenador.

Las configuraciones se probaron integrando el operador mix-opt en un algoritmo de Recocido Simulado. Los resultados obtenidos en media por cada configuración se almacenaron en una matriz en la que cada configuración tenía una componente asociada. La Figura 3.21 muestra la matriz en escala de colores, representando el azul los mejores valores (menor distancia al óptimo) y el rojo los peores. La mejor solución obtuvo una distancia recorrida de 160,03 m y la peor de 185,97 m, es decir, que la mejor solución disminuye aproximadamente en un 14% la distancia de la peor solución lo que es un valor importante para un campo tan pequeño como el considerado. La configuración que obtuvo los mejores valores fue $w_1 = 5, w_2 = 5, w_3 = 10, w_4 = 10, w_5 = 5, a = 5, l = 25\%$ y $r = cierto$. De la Figura 3.21 se extrae que en general es conveniente probar también con los segmentos invertidos ($r = cierto$), ya que en la figura hay mucho más azul oscuro en la parte asociada a esta opción. Asimismo, 25 % es la longitud máxima de segmento respecto a la longitud del vector que obtuvo mejores resultados mientras que el número de intentos más prometedor se situó en 5, ya que las columnas con este valor muestran por lo general un tono azul más oscuro que las de alrededor. Sobre los pesos, las configuraciones con $w_3 = 1$ y $w_4 = 1$ muestran peores resultados, como se puede observar en las líneas horizontales en azul más claro, especialmente apreciable para

$r = \text{cierto}$. Esto se debe a que estos valores son pequeños y sus operaciones asociadas son las operaciones de segmentos, por tanto, estas configuraciones no tienen capacidad suficiente de producir grandes cambios y dan lugar a una convergencia más lenta.

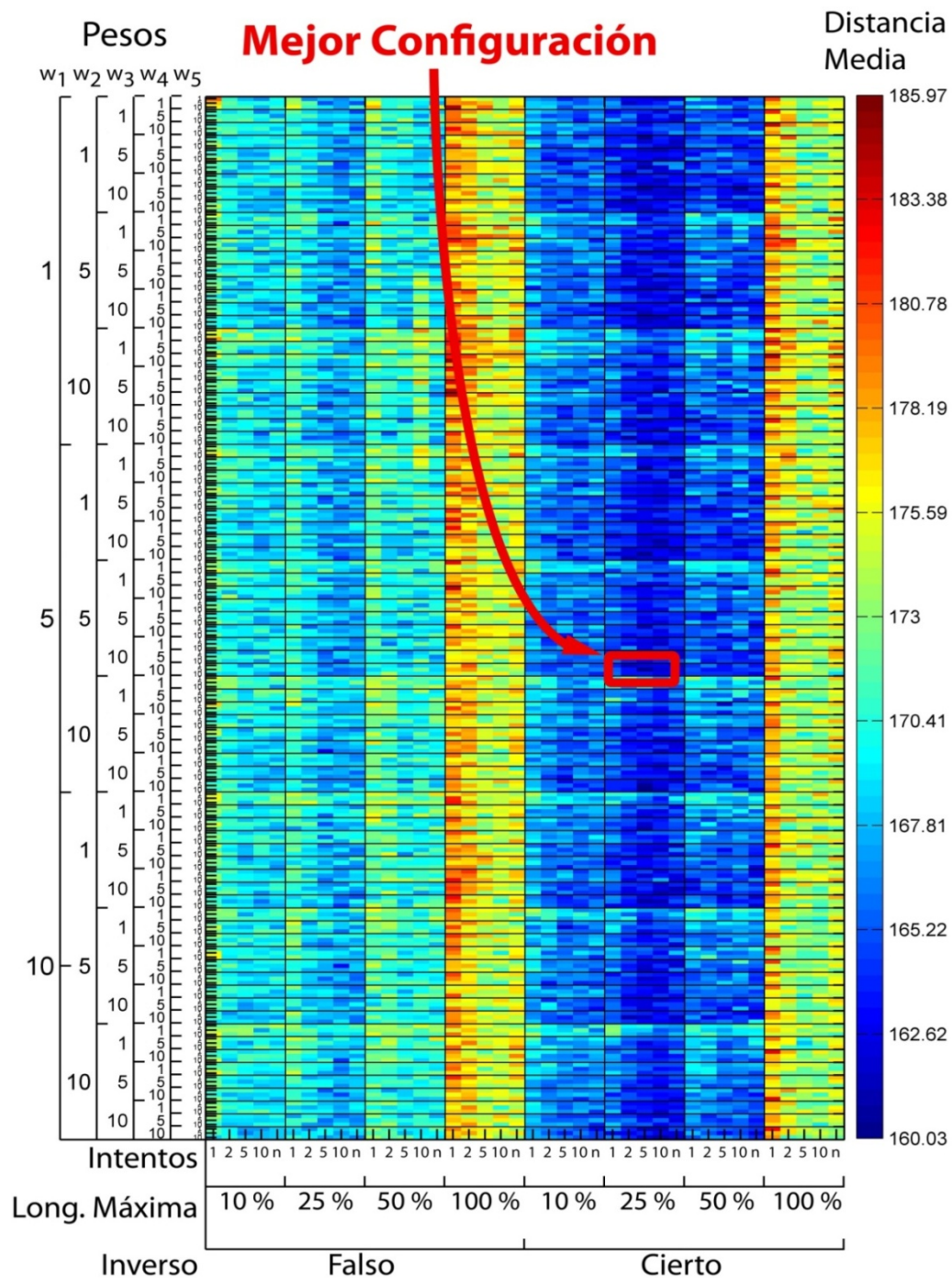


Figura 3.21. Valores objetivo en media (10 ejecuciones por configuración) para las 9.720 configuraciones consideradas del operador mix-opt

La mejor configuración encontrada es la que se ha establecido como configuración por defecto para el operador mix-opt, aunque siempre es posible variar los parámetros del operador dependiendo del problema. De hecho, con el nuevo operador se han mejorado los resultados variando moderadamente el valor de algunos parámetros tal como se muestra en la sección de resultados (sección 3.4.1), donde se han evaluado algunas configuraciones alternativas, también llamadas variantes. En definitiva la configuración por defecto debe entenderse como una orientación para evitar valores claramente malos, por ejemplo $a = 1$ o $l = 100$ %.

3.3.2. Configuración de los parámetros del método Recocido Simulado

Las variables del Recocido Simulado a configurar son la temperatura inicial T_0 , el factor de enfriamiento α , el tiempo de estancamiento antes de recalentar $t_{estancado}$ y el operador de vecindad, y, salvo el último operador, el resto de variables son muy dependientes del problema y no pueden configurarse como en el apartado anterior, muestreando y seleccionando los mejores valores. Por ejemplo, en el caso de la temperatura, hay que ajustarla de tal modo que permita cambios δE relativamente grandes cuando esté alta, para que sea posible transitar a zonas del espacio de búsqueda alejadas y así poder escapar de máximos/mínimos locales. Sin embargo, determinar si un cambio es grande depende totalmente del problema, ya que, por ejemplo, una diferencia de 10 unidades puede ser mínima en algunas optimizaciones mientras que muy grande en otras. Para resolver esta cuestión se utilizó la cota inferior del óptimo como medida del orden de magnitud de la optimización, de tal modo que se consideraron como cambios suficientemente grandes a aquellas diferencias δE que fuesen el doble de grandes que la cota inferior. En concreto se decidió usar una probabilidad de 0,5 de aceptación de estos cambios cuando la temperatura estuviese en su valor más alto. Como la probabilidad de aceptación $P(\delta E, T)$ del Recocido Simulado es de la forma $\exp\left(-\frac{\delta E}{T}\right)$ cuando $\delta E > 0$, la temperatura puede determinarse resolviendo la ecuación $\exp\left(-\frac{\delta E}{T}\right) = 0,5$, para los δE considerados como suficientemente grandes. En el caso del problema analizado durante el proceso de configuración, como la cota inferior era 124,75 m, la temperatura T_0 se fijó en 360 para que la probabilidad de aceptar un cambio de $2 \cdot 124,75$ unidades fuera 0,5 cuando la temperatura estuviera en su punto más alto, es decir, para $T = T_0$.

El factor de enfriamiento se configuró de tal modo que permitiera bajar la temperatura a 0 en pocos segundos, de este modo se consigue que solo puedan aceptarse soluciones a peor durante un breve periodo de tiempo y la búsqueda no se desboque por exceso de aleatoriedad. En concreto se usó el valor $\alpha = 0,95$, de este modo se consigue que en 100 configuraciones la temperatura se reduzca hasta un 0,6 % de su máximo valor (T_0), ya que $0,95^{100}$ es igual a 0,006.

El tiempo de estancamiento es muy dependiente del tamaño del problema, siendo necesario más tiempo cuanto mayor es la complejidad, por lo que se fijó en función del número de variables de decisión del problema, en concreto, para este caso como un cuarto del número de pistas del campo, es decir, 10 segundos.

Como operador de vecindad se empleó el mix-opt para la parte de la solución que contiene el vector permutación mientras que para el vector de reabastecimientos se probaron las dos alternativas explicadas en la sección 3.2.5.2, es decir, la inversión

directa y la acción aleatoria. Asimismo, también se estudió la opción de reparar las soluciones inviables que pueden parecer tras modificar el vector de reabastecimientos siguiendo las directrices explicadas en la sección 3.2.2.3. Ninguna de estas últimas opciones son específicas del Recocido Simulado y pueden aplicarse a los otros métodos de optimización considerados, sin embargo se analizaron con el Recocido Simulado por ser el método que requiere la comprobación de un menor número de configuraciones. En consecuencia, las decisiones para los otros métodos de optimización como si usar o no la reparación de soluciones o el tipo de operador de reabastecimientos se supeditaron a los resultados obtenidos con el Recocido Simulado. La Tabla 3.9 detalla los valores muestreados para cada variable.

Tabla 3.9. Valores de configuración probados para el Recocido Simulado

Variantes	Valores probados
Reparar solución	{NO, SI}
Operador de reabastecimientos	{ <i>inversionBinaria</i> , <i>accionAleatoria</i> }

La Figura 3.22 muestra las convergencias de las cuatro configuraciones evaluadas. Como se observa, la que mejores resultados presenta es la que evita la reparación de soluciones y utiliza un operador de acción aleatoria para el vector de reabastecimientos. A la vista de los resultados se optó, tal como se verá más adelante, por no utilizar la reparación de soluciones ni la inversión binaria en los métodos de optimización basados en Algoritmos Genéticos y NSGA-II.

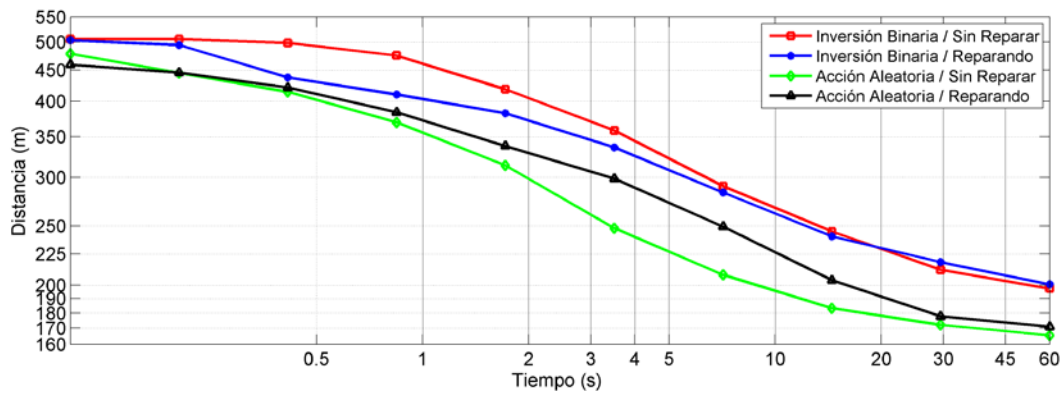


Figura 3.22. Valores objetivo medios, obtenidos en el proceso de configuración del Recocido Simulado

3.3.3. Configuración de los parámetros del Algoritmo Genético

En el caso del Algoritmo Genético se estudiaron las siguientes variables de configuración (véase Tabla 3.10): el tamaño de población o número de individuos (n_p), el tipo de elitismo (e), la probabilidad de cruce (p_c) y la probabilidad de mutación (p_m). El tipo de selección se excluyó del estudio y se optó por usar siempre la selección por ruleta por ser una opción que aparece con frecuencia en la literatura, de este modo se reduce el número total de configuraciones a evaluar.

Tabla 3.10. Valores de configuración examinados para el Algoritmo Genético

Variables	Valores probados	Descripción
$n_p \in N_p$	$N_p = \{2, 4, 8, 10, 20, 30, 40\}$	Tamaño de población
$e \in E$	$E = \{Ninguno, Individual, Poblacional\}$	Tipo de elitismo
$p_c \in P_c$	$P_c = \{0, 1, 0,25, 0,5, 0,75, 1\}$	Probabilidad de cruce
$p_m \in P_m$	$P_m = \{0, 1, 0,25, 0,5, 0,75, 1\}$	Probabilidad de mutación

En total se probaron 525 configuraciones, es decir, $|N_p| \cdot |E| \cdot |P_c| \cdot |P_m|$ y se realizaron 10 ejecuciones del algoritmo para cada configuración. La Figura 3.23 muestra el valor medio obtenido en cada caso. La mejor configuración mostraba el siguiente conjunto de valores: $n_p = 2$, $e = Poblacional$, $p_c = 0,25$ y $p_m = 0,25$, y daba lugar a una solución con una distancia recorrida de 165,19 m. Sin embargo, debido a que la probabilidad de cruce y el tamaño de población son muy bajos, se optó por seleccionar como valores por defecto los de una configuración que daba lugar a una solución ligeramente peor con un recorrido de 166,239 m. En este caso los valores de configuración son: $n_p = 20$, $e = Poblacional$, $p_c = 1$, $p_m = 0,25$. Tanto la mejor configuración como la finalmente seleccionada aparecen indicadas en la Figura 3.23.

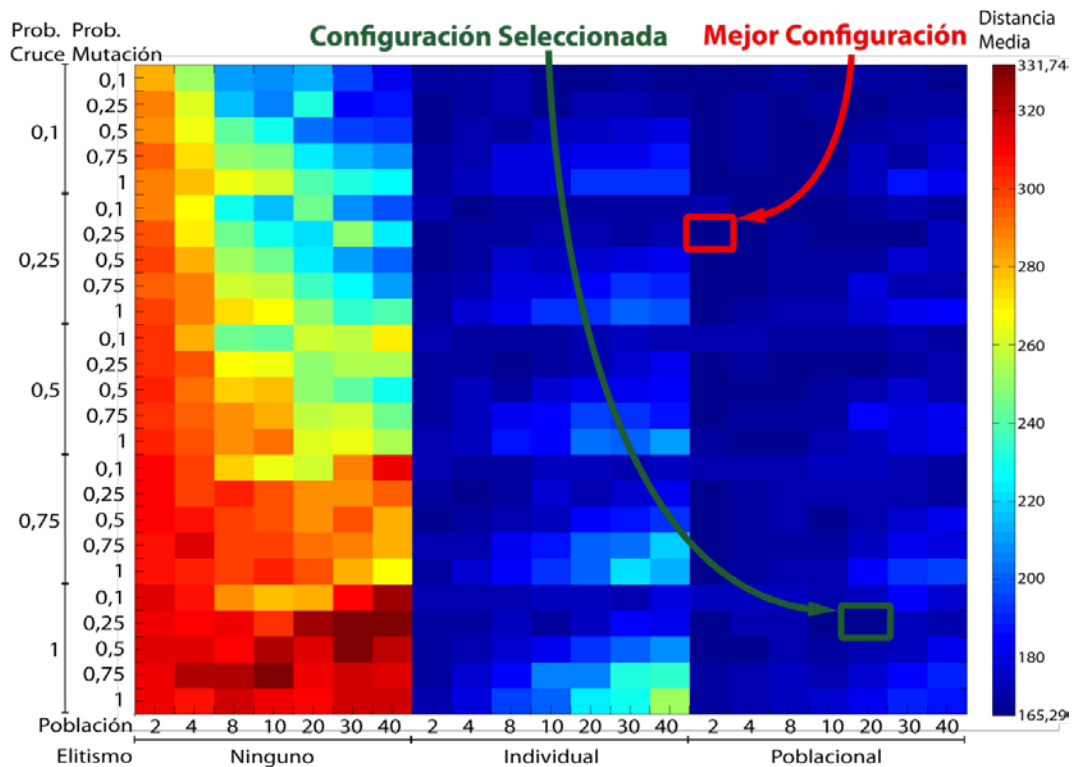


Figura 3.23. Resultados medios obtenidos en el proceso de configuración del Algoritmo Genético considerado

A la vista de la figura anterior se pueden extraer conclusiones como que no usar elitismo perjudica la convergencia, asimismo, en este caso las poblaciones grandes no garantizan mejores resultados cuando se utiliza elitismo, incluso se observa un leve empeoramiento para poblaciones superiores a 20 individuos. Por último, las configuraciones con alta probabilidad de mutación (0,75 y 1) presentan valores ligeramente peores, como era de esperar, especialmente cuanto mayor es la probabilidad de cruce y en aquellas configuraciones con elitismo individual. No obstante, en general todas las configuraciones muestran un buen rendimiento excepto las que no usan elitismo, por lo que queda claro que en este tipo de problemas conviene utilizar algún tipo de elitismo en la etapa de reemplazo.

3.3.4. Configuración de los parámetros del NSGA-II

Como se comentó, en el caso del NSGA-II se descartó la distancia como criterio de optimización para emplear dos costes claramente contrapuestos: el coste por insumos y el tiempo requerido para realizar la tarea. Para ello se usaron los valores de la Tabla 3.11 extraídos de la sección de costes (3.2.3.1) con alguna modificación, concretamente en el coste de la dosis de pulverización para adaptarlo a un precio más real, esto es, a los 90 €/ha que cuesta el Rimsulfuron (Andújar, Ribeiro, Fernández-Quintanilla, & Dorado, 2011). Como el mapa de malas hierbas que se consideró dividía el campo en celdas de 75 x 75 cm, una dosis de herbicida sobre esa superficie cuesta $0,51 \cdot 10^{-2}$ €

Tabla 3.11. Costes para la flota en el proceso de configuración del NSGA-II

Arranque		Desp. Inter-pista		Desp. Intra-pista		Pulverización		Reabast.	
€	t (s)	€(10^{-3})	t (s)	€(10^{-3})	t (s)	€(10^{-2})	t (s)	€	t (s)
0,2	0	0,96	0,36	0,96	0,6	0,51	0	0	120

Las variables de configuración del NSGA-II son las mismas que las de los Algoritmos Genéticos, con la excepción del tipo de elitismo que en este caso no se utiliza y en su lugar se considera el tamaño (número de individuos) de la nueva población (véase Tabla 3.12).

Tabla 3.12. Valores analizados para cada variable del método NSGA-II

Variables	Valores probados	Descripción
$n_p \in N_p$	$N_p = \{3,5,7,10,20,30\}$	Tamaño de población de trabajo (progenitora)
$n_h \in N_h$	$N_h = \{25 \%, 50 \%, 100 \%, 200 \%\}$	Tamaño de la nueva población respecto a la progenitora
$p_c \in P_c$	$P_c = \{0,1,0,25,0,5,0,75,1\}$	Probabilidad de cruce
$p_m \in P_m$	$P_m = \{0,1,0,25,0,5,0,75,1\}$	Probabilidad de mutación

En total se probaron 600 configuraciones, es decir, $|N_p| \cdot |N_h| \cdot |P_c| \cdot |P_m|$, con 10 ejecuciones por configuración. La Figura 3.24 muestra los resultados obtenidos en media por cada configuración.

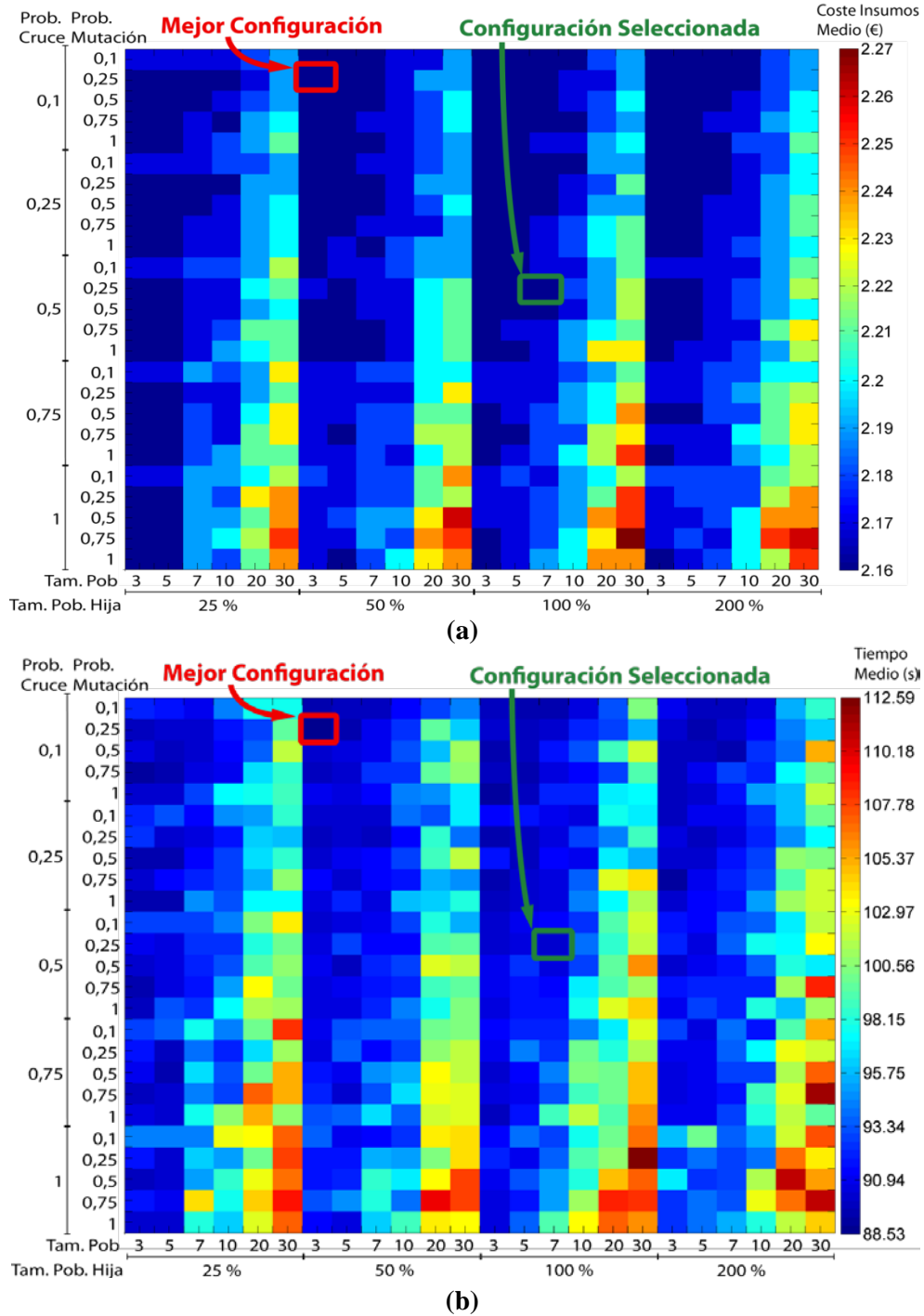


Figura 3.24. Valores objetivo medios normalizados, obtenidos en el proceso de configuración del NSGA-II para los criterios (a) coste por insumos y (b) tiempo de tarea.

Al ser un algoritmo multi-objetivo, el proceso calcula a la vez los valores de las configuraciones para los dos criterios establecidos. Para ambos criterios la mejor solución se dio con la configuración siguiente: $n_p = 3$, $n_h = 50\%$, $p_c = 0,1$ y $p_m = 0,25$. La mejor solución/plan tiene un coste en insumos de 2,16 € y en tiempo de 88,53 s. Ahora bien, al igual que con el método basado en Algoritmos Genéticos, se optó por escoger otra configuración de entre las mejores con un tamaño de población mayor para disponer de mayor variedad de soluciones en problemas más complejos con mayores espacios de búsqueda. En concreto, la configuración que se eligió finalmente fue $n_p = 7$, $n_h = 100\%$, $p_c = 0,5$ y $p_m = 0,25$, que suministraba una solución con unos valores medios ligeramente superiores, 2,16 € y 90,23 s.

A la vista de la Figura 3.24 se puede concluir que el tamaño de la nueva población respecto a la población progenitora no tiene un impacto relevante en las configuraciones, sin embargo, sí se observa que el rendimiento decrece al aumentar el tamaño de la población y la probabilidad de cruce. Aún así, no es conveniente reducir en exceso el tamaño de la población puesto que entonces se reduciría también el frente, y con él el número de soluciones intermedias, las cuales pueden resultar interesantes según la relevancia que se le quiera dar a cada uno de los criterios. Las soluciones intermedias son aquellas que están en el primer frente pero no son las mejores para un único criterio, es decir, no están en los extremos del frente. Es conveniente no tener una población excesivamente pequeña para que haya cierto número de ellas y con diferentes grados de optimización según los objetivos, es decir, más optimizadas para el primer criterio, más optimizadas para el segundo, etc.

La Figura 3.25 muestra las soluciones intermedias para un frente de 20 individuos optimizado en coste por insumos y tiempo.

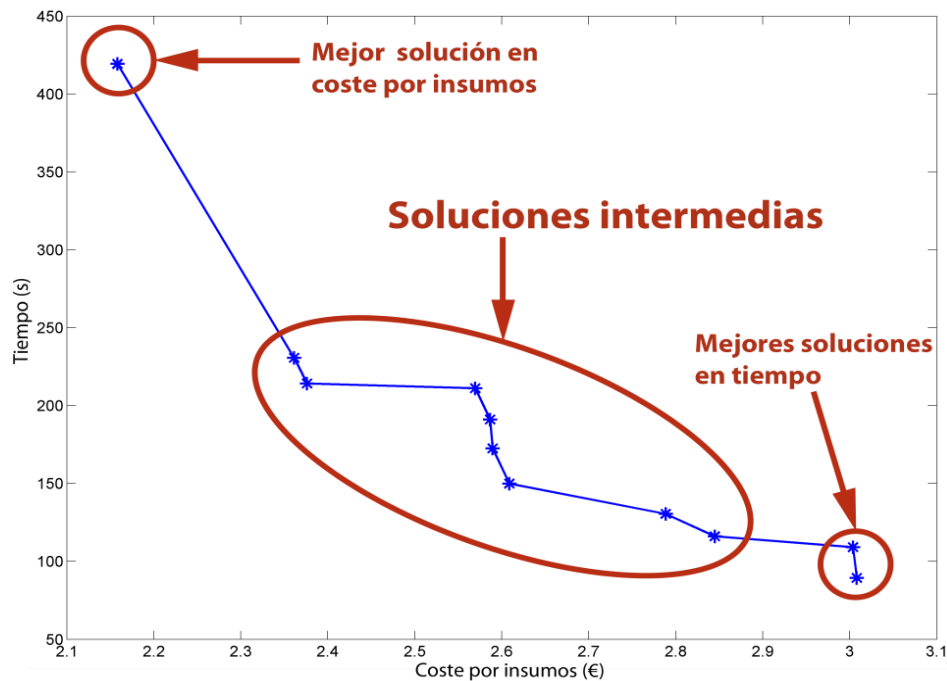


Figura 3.25. Población de soluciones optimizadas en coste por insumos y tiempo simultáneamente

Como se puede ver, con dos objetivos la población se divide en tres sub-conjuntos, el primero con las soluciones que minimizan el primer criterio, el segundo con las soluciones que minimizan el segundo objetivo, y el tercero con las soluciones que minimizan criterios conjuntamente.

3.4. Resultados

Una vez encontradas unas configuraciones adecuadas tanto para el operador mix-opt como para los métodos de optimización (Recocido Simulado, Algoritmo Genético y NSGA-II), el siguiente paso consiste en comparar el nuevo operador propuesto con los operadores que se pueden encontrar en la literatura así como comparar los tres métodos de optimización entre sí. El primer estudio comparativo tiene la finalidad de determinar si el operador propuesto supera en rendimiento a los operadores que se han estado utilizando hasta el momento. El segundo estudio pretende determinar cuál de los tres métodos es capaz de obtener mejores resultados en la resolución de problemas habituales e ilustrativos de planificación agrícola en los que sea fácil analizar la coherencia de la solución proporcionada por el método de optimización. El propósito es que se pueda verificar fácilmente que las soluciones obtenidas son buenas y que, por tanto, el planteamiento propuesto en este trabajo de investigación tiene validez por ser útil en numerosos escenarios reales. Por último, se resolverán algunos problemas de planificación propuestos en otros trabajos de investigación y que se recogen en la literatura para comprobar el rendimiento de la aproximación propuesta en esta tesis.

Tanto los operadores como los métodos de optimización necesarios para este estudio comparativo se han implementado en Matlab. A pesar de que el código generado en Matlab se considera lento frente al mismo código escrito en lenguajes como pueden ser C o C++, se optó por usar la plataforma Matlab por disponer de una toolbox de optimización, lo que permitía partir de un software muy depurado. El problema de la lentitud de Matlab no afecta al estudio comparativo porque tanto el operador mix-opt como los métodos estarán en igualdad de condiciones. En el caso de la comparativa con otras aproximaciones, el objetivo fue demostrar que es posible alcanzar mejores soluciones en un tiempo razonable, aunque sea mayor que el que se podría haber conseguido desarrollando los métodos en otro lenguaje. Por último, todas las pruebas se ejecutaron en un ordenador Intel Core i5-2500 CPU @ 3.30 GHz con 4 GB de RAM.

Algunos de estos resultados así como las principales aportaciones del método de planificación propuesto fueron publicados en (Conesa-Muñoz, Ribeiro, Andujar, Fernandez-Quintanilla, & Dorado, 2012) y (Conesa-Munoz, Bengochea-Guevara, Andujar, & Ribeiro, 2015).

3.4.1. El operador mix-opt versus los operadores de la literatura

Aunque el operador mix-opt se planteó para encontrar el mejor orden en el que recorrer las pistas de un cultivo, también puede emplearse para resolver problemas CVRPs clásicos, entendiendo en ese caso que los elementos que se recombinan son clientes en lugar de pistas y no permitiendo reabastecimientos. Se decidió aprovechar que existen bibliotecas de problemas CVRP que proporcionan la descripción de problemas y las soluciones óptimas asociadas para comparar el operador mix-opt con los operadores

que aparecen en la literatura, descritos en la sección 3.2.5.2, ya que la mayoría de estos se desarrollaron para resolver problemas TSP y CVRP. Además, el estudio comparativo incluye pruebas con planificaciones agrícolas. Para ambos casos la estrategia seguida fue similar: 1) integrar diferentes operadores en diferentes Recocidos Simulados de tal modo que cada uno de ellos construyera los estados vecinos usando siempre el mismo operador y 2) comparar los resultados obtenidos.

3.4.1.1. Rendimiento en problemas CVRP

En este análisis se utilizó la biblioteca de (Christofides & Eilon's Instances, 2015), un conjunto de problemas muy usados como referencia para comparar algoritmos que resuelven el problema CVRP. En total contiene 15 problemas con sus respectivas soluciones óptimas. Se utilizaron un total de 10 problemas de los 15 disponibles, descartando el resto porque no presentaban diferencias sustanciales en el número de vehículos y/o clientes con los problemas seleccionados. En los problemas escogidos el rango de clientes estaba entre 22 y 101 y el de vehículos en la flota entre 4 y 14.

Cada problema se testeó 100 veces por cada operador empleando un Recocido Simulado como algoritmo de optimización, siendo la duración de cada ensayo de 1 hora. El criterio que se optimizó en todos los casos fue la distancia recorrida utilizando una métrica euclídea. El valor objetivo de cada operador para cada problema se obtuvo haciendo la media de las 100 ejecuciones y se expresó en términos de *gap* tal y como se describió en la sección 3.2.3.2. En este caso como la diferencia entre la distancia recorrida en media utilizando las mejores soluciones de cada ejecución y el valor óptimo disponible en la biblioteca, en términos porcentuales del óptimo.

Para impedir la aparición de soluciones con reabastecimientos –no considerados en los problemas de la biblioteca empleada–, se ajustó el coste por reabastecimiento a $+\infty$.

Además de los operadores de la literatura, se probaron también algunas variantes del operador mix-opt con objeto de determinar los límites de dependencia del operador de la configuración establecida por defecto y la dependencia de cada una de las operaciones elementales que lo componen. La Tabla 3.13 muestra los *gaps* obtenidos usando en exclusiva cada operador para cada problema. Los problemas evaluados se han identificado siguiendo la notación $E-nX-kY$, siendo X el número de clientes del problema e Y el de vehículos.

Las variantes resultan de descartar algunas de las operaciones de las que está compuesto el mix-opt, esto puede hacerse configurando el peso w_i asociado a estas operaciones a 0, para que nunca sean escogidas. Por ejemplo, en el caso de la variante *mix-opt sin intercambios*, se ajustaron a 0 los pesos de las operaciones elementales *intercambio unitario* e *intercambio de segmentos*, o en el caso de la variante *mix-opt sin operaciones unitarias* se ajustaron a 0 las operaciones *desplazamiento unitario* e *intercambio unitario*.

La Figura 3.26 muestra las posiciones en la que quedó cada operador en la comparación para cada problema según sus resultados en media.

Tabla 3.13. Resultados en media obtenidos por cada operador para cada uno de los 10 problemas CVRP analizados. Las filas en gris resaltan los resultados del operador mix-opt y sus variantes.

Problemas Operadores	E-n22-k4	E-n23-k3	E-n30-k3	E-n33-k4	E-n51-k5	E-n76-k7	E-n76-k10	E-n76-k14	E-n101-k8	E-n101-k14
Mix-opt	0,004	0	0,008	0,004	0,031	0,045	0,059	0,063	0,069	0,097
Simple Inversion	0,024	0,001	0,042	0,022	0,088	0,056	0,107	0,102	0,066	0,088
Cut Inversion	0,022	0	0,028	0,029	0,096	0,057	0,114	0,112	0,074	0,101
LSM	0,424	0,074	0,295	0,137	0,618	0,147	0,795	0,757	0,186	0,224
Or-opt	0,129	0	0,143	0,053	0,296	0,082	0,278	0,259	0,112	0,149
Relocate	0,273	0,042	0,163	0,069	0,427	0,122	0,711	0,717	0,154	0,18
Scramble	0,097	0,108	0,184	0,092	0,353	0,377	0,368	0,286	0,47	0,392
Swap	0,157	0,139	0,251	0,101	0,294	0,326	0,308	0,246	0,393	0,346
Mix-opt sin despl. de segmento inter-veh.	0,002	0	0,008	0,005	0,035	0,047	0,06	0,061	0,071	0,1
Mix-opt sin desplazamientos	0,001	0	0,007	0,005	0,033	0,055	0,07	0,069	0,092	0,111
Mix-opt sin operaciones con segmentos ¹	0,11	0,034	0,113	0,052	0,207	0,119	0,214	0,176	0,145	0,133
Mix-opt sin desplazamientos de segmento	0,002	0	0,007	0,005	0,03	0,043	0,058	0,059	0,068	0,091
Mix-opt sin intercambio de segmentos	0,012	0	0,01	0,008	0,062	0,037	0,081	0,081	0,057	0,085
Mix-opt sin operaciones unitarias ¹	0,002	0	0,007	0,004	0,03	0,053	0,068	0,074	0,092	0,115
Mix-opt sin desplazamiento unitario	0,002	0	0,007	0,004	0,035	0,05	0,061	0,066	0,08	0,099
Mix-opt sin intercambio unitario	0,003	0	0,007	0,004	0,04	0,046	0,066	0,07	0,071	0,103
Mix-opt sin intercambios	0,011	0	0,011	0,007	0,068	0,042	0,104	0,11	0,064	0,095

¹ Operaciones indica tanto los desplazamientos como los intercambios.

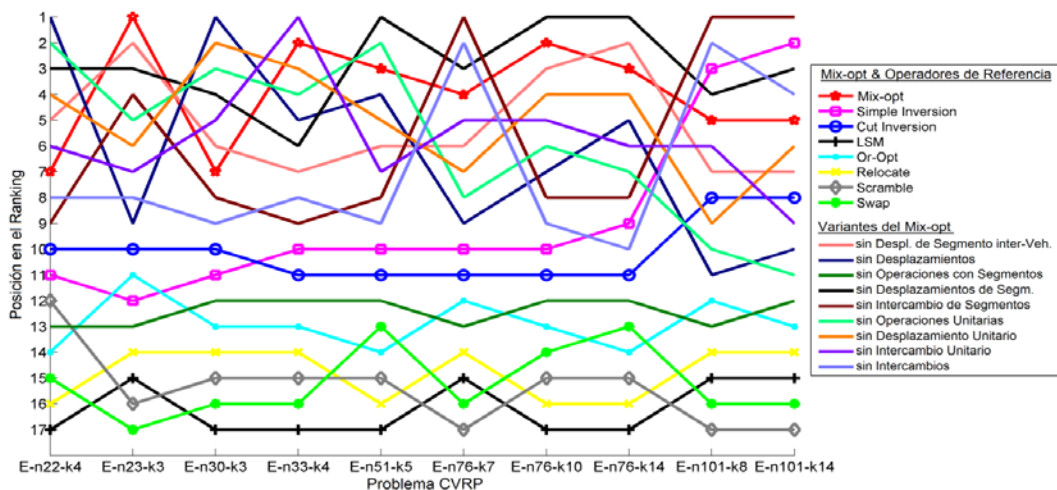


Figura 3.26. Clasificación de los operadores para cada problema CVRP según sus resultados medios.

La Tabla 3.14 muestra los resultados en media sin desglosar por problema, además presenta la mejora en términos porcentuales con respecto al operador de la literatura (*simple inversion*) que obtuvo los mejores resultados. Tanto los resultados del operador mix-opt como los de sus variantes se resaltan en gris. El operador mix-opt y sus variantes ocupan 9 de los 10 primeras posiciones; la única variante superada por alguno de los operadores de la literatura es la versión del operador mix-opt que no contiene operaciones con segmentos y esto la limita de modo que no puede llevar a cabo grandes cambios en pocos pasos.

La configuración por defecto para el operador mix-opt, tal como se ajustó en la sección de configuración, ocupa la segunda posición. Este comportamiento está dentro de lo previsible ya que el tiempo utilizado en el proceso de configuración fue pequeño y, por tanto, tal como se comentó en la sección, la configuración obtenida es buena en general aunque no tiene porque ser la mejor para todos los casos.

Dado que las nueve primeras posiciones están ocupadas por versiones del nuevo operador, se puede concluir que, en términos generales, su rendimiento claramente mejora el de los operadores ya existentes.

Tabla 3.14. Resultados en media ordenados y mejoras respecto al operador de la literatura que obtuvo mejores resultados (*simple inversion*). En gris el mix-opt y sus variantes.

Operador	Gap medio	Mejora respecto al operador referencia (%) (<i>simple inversion</i>)
Mix-opt sin desplazamiento de segmentos	0,0362	39,27
Mix-opt	0,0379	36,42
Mix-opt sin despl. de segmentos inter-vehículo	0,0389	34,70
Mix-opt sin desplazamiento unitario	0,0403	32,38
Mix-opt sin intercambio unitario	0,0409	31,39
Mix-opt sin intercambio de segmentos	0,0432	27,56
Mix-opt sin desplazamientos	0,0441	25,98
Mix-opt sin operaciones unitarias	0,0444	25,46
Mix-opt sin intercambios	0,0511	14,29
Simple Inversion	0,0596	0
Cut Inversion	0,0632	-6,06*
Mix-opt sin operaciones con segmentos	0,1302	-118,55*
Or-Opt	0,1502	-152,03*
Swap	0,256	-329,58*
Scramble	0,2728	-357,83*
Relocate	0,2858	-379,59*
LSM	0,3655	-513,41*

*Los porcentajes negativos indican peores resultados que los del operador referencia.

La Figura 3.27 muestra para cada problema las diferencias entre los resultados medios obtenidos por cada operador y 0, el valor mínimo posible. La mayoría de las variantes del mix-opt muestran un rendimiento muy similar, hasta tal punto que es imposible diferenciarlas. Tan solo el operador *simple inversion* y el *cut inversion* están relativamente cerca del operador propuesto.

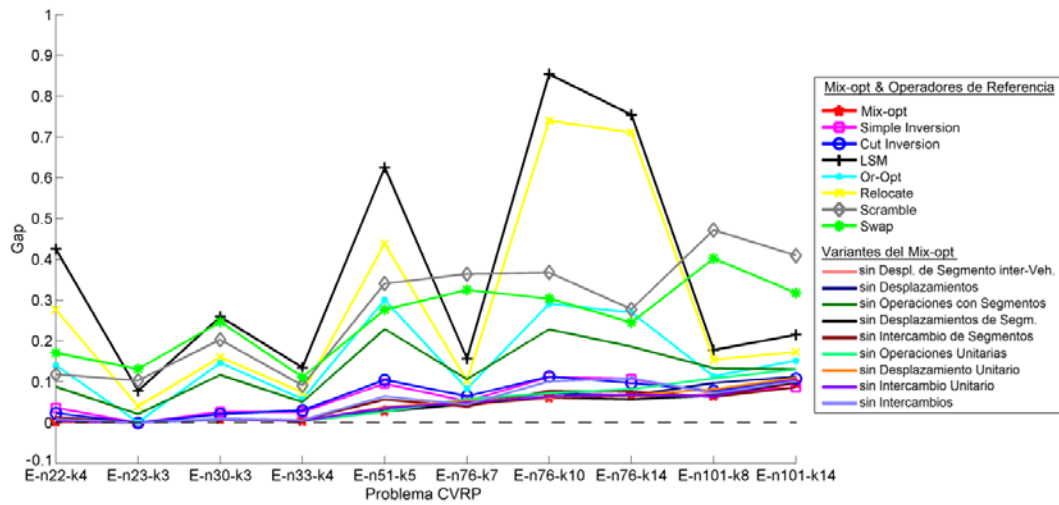


Figura 3.27. Gaps para cada operador en cada problema CVRP.

El mismo proceso se llevó a cabo durante 1, 7,5, 15 y 30 minutos para comprobar el tiempo que necesita el nuevo operador para imponerse a los demás y, como se observa, el operador mix-opt y sus variantes superaron al resto de operadores (véase la Figura 3.28 basada en los datos de la Tabla 3.15) para todos los tiempos excepto para 1 minuto.

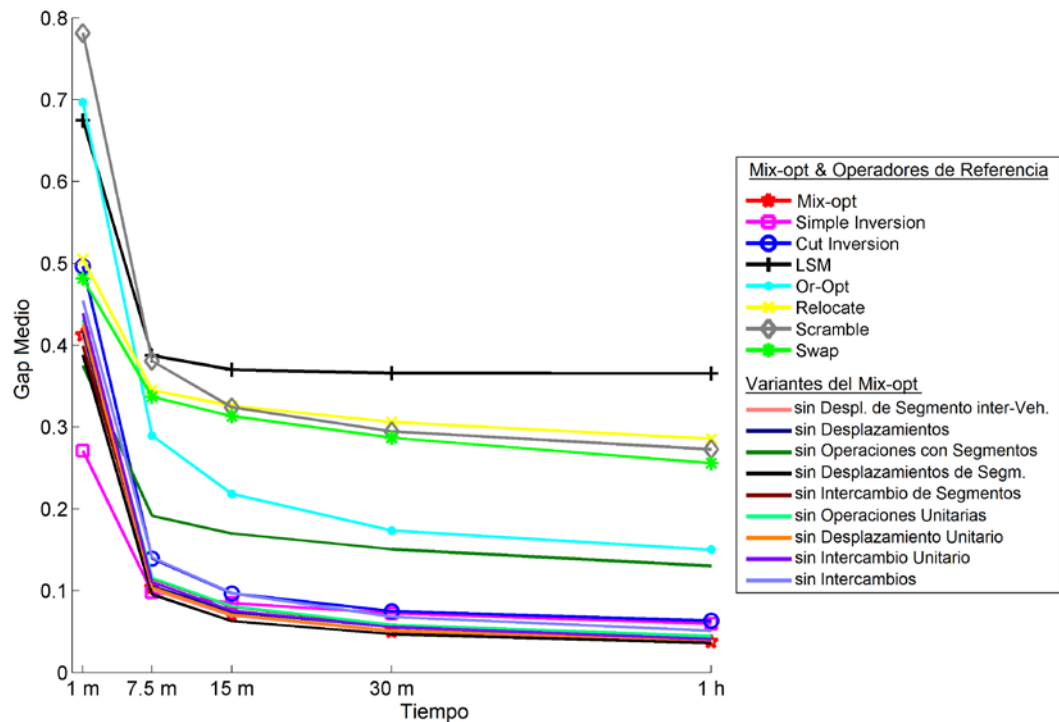


Figura 3.28. Gaps medios de los operadores para 1, 7,5, 15, 30 y 60 minutos.

Tabla 3.15. *Gaps* medios ordenados y mejoras para cada operador respecto al mejor operador de la literatura para 60, 30, 15, 7,5 y 1 minutos. En gris se resaltan los resultados del operador mix-opt y sus variantes.

60 Minutes			30 Minutes			15 Minutes			7,5 Minutes			1 Minutes		
Operator	Gap (%)	Operator	Gap (%)	Operator	Gap (%)	Operator	Gap (%)	Operator	Gap (%)	Operator	Gap (%)	Operator	Gap (%)	Operator
Mix-opt sin despl. de segmentos	0,036	39,3	Mix-opt sin despl. de segmentos	0,047	35,1	Mix-opt sin despl. de segmentos	0,063	25,9	Mix-opt sin despl. de segmentos	0,096	2,6	Simple Inversion*	0,271	0
Mix-opt	0,038	36,4	Mix-opt sin despl. de segm. inter-veh.	0,05	31	Mix-opt sin despl. de segmento inter-veh.	0,07	18	Simple Inversion*	0,098	0	Mix-opt sin ops. con segmentos	0,375	-38,3
Mix-opt sin despl. de segm. inter-veh.	0,039	34,7	Mix-opt	0,05	30,9	Mix-opt	0,07	17,5	Mix-opt sin despl. de segm. inter-veh.	0,102	-3,7	Mix-opt sin despl. de segmentos	0,388	-43,2
Mix-opt sin despl. unitario	0,04	32,4	Mix-opt sin despl. unitario	0,051	29,3	Mix-opt sin despl. unitario	0,071	16,3	Mix-opt	0,104	-5,9	Mix-opt sin intercambio de segmentos	0,399	-47,3
Mix-opt sin intercambio unitario	0,041	31,4	Mix-opt sin intercambio unitario	0,056	23,4	Mix-opt sin desplazamientos	0,074	12,8	Mix-opt sin despl. unitario	0,104	-5,9	Mix-opt sin despl. de seg. inter-veh.	0,411	-51,4
Mix-opt sin intercambio de segmentos	0,043	27,6	Mix-opt sin desplazamientos	0,056	23,1	Mix-opt sin intercambio unitario	0,076	10,8	Mix-opt sin desplazamientos	0,105	-7,2	Mix-opt	0,413	-52,1
Mix-opt sin desplazamientos	0,044	26	Mix-opt sin intercambio de segmentos	0,057	21	Mix-opt sin intercambio de segmentos	0,081	5	Mix-opt sin intercambio unitario	0,11	-12,6	Mix-opt sin desplazamientos	0,418	-54,2
Mix-opt sin operaciones unitarias	0,044	25,5	Mix-opt sin operaciones unitarias	0,058	19,5	Mix-opt sin operaciones unitarias	0,081	4,6	Mix-opt sin intercambio de segmentos	0,115	-17,3	Mix-opt sin despl. unitario	0,425	-56,8
Mix-opt sin intercambios	0,051	14,3	Mix-opt sin intercambios	0,068	6	Simple Inversion*	0,085	0	Mix-opt sin operaciones unitarias	0,116	-18,1	Mix-opt sin operaciones unitarias	0,431	-59
Simple Inversion*	0,06	0	Simple Inversion*	0,073	0	Cut Inversion	0,097	-14	Cut Inversion	0,139	-42,1	Mix-opt sin intercambio unitario	0,439	-61,8
Cut Inversion	0,063	-6,1	Cut Inversion	0,075	-3,5	Mix-opt sin intercambios	0,097	-14,3	Mix-opt sin intercambios	0,141	-43,3	Mix-opt sin intercambios	0,454	-67,6
Mix-opt sin ops. con segmentos	0,13	-118,5	Mix-opt sin ops. con segmentos	0,151	-108	Mix-opt sin ops. con segmentos	0,17	-100,1	Mix-opt sin ops. con segmentos	0,192	-95,6	Swap	0,481	-77,5
Or-Opt	0,15	-152	Or-Opt	0,174	-139,4	Or-Opt	0,218	-157,2	Or-Opt	0,29	-195,6	Cut Inversion	0,496	-83
Swap	0,256	-329,6	Swap	0,287	-295,9	Swap	0,313	-269,1	Swap	0,337	-243,7	Relocate	0,505	-86,3
Scramble	0,273	-357,8	Scramble	0,295	-306,7	Relocate	0,324	-282	Relocate	0,345	-252,1	LSM	0,675	-148,8
Relocate	0,286	-379,6	Relocate	0,306	-322,6	Scramble	0,326	-284,2	Scramble	0,381	-288,4	Or-Opt	0,697	-157
LSM	0,366	-513,4	LSM	0,366	-404,8	LSM	0,37	-335,9	LSM	0,388	-295,4	Scramble	0,782	-188,3

*Mejor operador de los extraídos de la literatura

**Los porcentajes negativos indican que los resultados son peores que los del operador de referencia

Los resultados para el primer minuto se explican por la ausencia de etapas tardías de exploración en las que cambios pequeños y precisos son habitualmente necesarios para mejorar las soluciones. En un solo minuto, la búsqueda se desarrolla solo en las primeras etapas en las que los cambios grandes, tales como los producidos por el operador *simple inversion*, son favorables, mientras que los cambios del operador mix-opt tienen un amplio espectro, es decir, están pensados para cubrir todo tipo de etapas del proceso de búsqueda.

3.4.1.2. Rendimiento en problemas agrícolas

Tras confirmar el buen funcionamiento del mix-opt en problemas CVRP clásicos, se llevó a cabo un estudio adicional para analizar el rendimiento del operador en un problema relacionado con el marco de actuación del trabajo de investigación presentado en esta tesis. En esta ocasión, el Recocido Simulado se usó para planificar un campo de 200 pistas mediante una flota de 5 vehículos homogéneos. El mínimo radio de giro de los vehículos se fijó en 3 m y el ancho de pista en 2 m. Como los campos se siembran de tal modo que las líneas de cultivo y por ende las pistas están orientadas siguiendo la dirección del lado más largo –de esta manera hay menos pistas aunque más largas y se necesitan menos giros en cabecera–, los campos de cultivo son por lo general como mínimo tan largos como anchos. Si en aras de claridad se supone una forma rectangular con 200 pistas de 2 m de ancho, el campo más pequeño que se puede formar tiene 16 ha, es decir, $(2 \cdot 200)^2 m^2$, el cual es un buen representante de un campo de cultivo extensivo real.

Para este estudio comparativo se ejecutaron 50 ensayos y cada uno de ellos duró 1 día. Se eligió un tiempo suficiente para garantizar que el proceso de búsqueda pasara por todas las etapas de convergencia. Al igual que en los ensayos para los problemas CVRP, el criterio de optimización considerado fue la distancia entre pistas, la distancia dentro del cultivo no se tuvo en cuenta por ser la misma para cualquier solución ya que por construcción cada pista tiene que ser recorrida exactamente una vez.

La Figura 3.29 muestra los resultados obtenidos según el tiempo de ejecución. Ambos ejes usan una escala logarítmica para apreciar mejor la magnitud de la mejora a lo largo del tiempo. Como el tiempo de ejecución se estableció en un valor elevado, 1 día, el tiempo máximo de estancamiento antes de recalentar se fijó en 20 minutos.

La Tabla 3.16 muestra los resultados obtenidos en media por cada operador para las 50 ejecuciones tras un día de optimización. En este caso, como se desconocía el óptimo, se calculó una cota inferior (1.875,53 m) y una superior (43.584,00 m) mediante las ecuaciones (3.18) y (3.20), y los *gaps* que se muestran se estimaron con respecto a la cota mínima. La tabla también muestra la mejora en relación al operador de la literatura que obtuvo los mejores resultados (*simple inversion*). Al igual que sucedió con los problemas CVRP clásicos, el operador mix-opt y sus variantes superan con creces a todos los operadores existentes. Únicamente, una vez más, la variante sin operaciones con segmentos fue incapaz de superar al resto de operadores. Los resultados fueron incluso mejores que los obtenidos para los problemas CVRP debido al mayor tiempo de ejecución, aunque a partir de las 3 horas el operador mix-opt y sus variantes mostraban signos de estancamiento no presentando apenas mejoras.

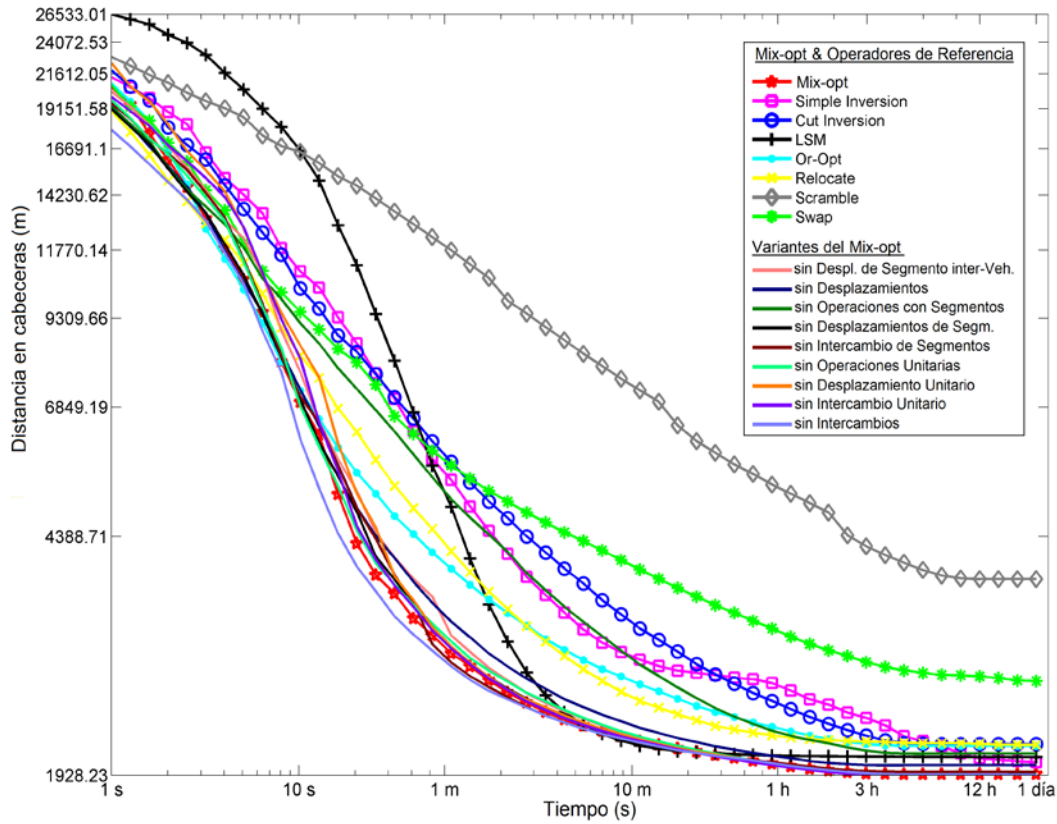


Figura 3.29. Convergencia de los operadores de la literatura, el operador mix-opt y sus variantes en un problema de planificación agrícola. Ambos ejes están en escala logarítmica.

La variante del operador *mix-opt sin desplazamiento de segmentos entre vehículos* obtuvo los mejores resultados. Esto se debe a que el número de vehículos era muy inferior al de pistas y por ello el grueso del trabajo consiste en encontrar el mejor orden de transición más que en distribuir las pistas entre las unidades móviles, siendo esto último precisamente a lo que se dedica la operación *de segmentos entre vehículos*. Como esta variante no utiliza esa operación, pudo dedicar más tiempo a trabajar en determinar el orden óptimo.

Por último, el operador mix-opt y su mejor variante superaron al mejor operador de referencia (*simple inversion*) en 56,64 % y 62,07 % respectivamente. Además, aunque las pruebas se lanzaran durante 1 día para garantizar que el proceso de convergencia pasara por todas las etapas, como se puede comprobar, no es necesario esperar horas para obtener grandes mejoras. A partir de los 10 minutos, muchos operadores ya están en valores cercanos a los 2.100 m, que no están muy alejados del óptimo si tenemos en cuenta que la cota inferior se estimó en 1.875,53 m, siendo este una cota inferior no alcanzable calculada mediante la ecuación (3.24).

Tabla 3.16. Resultados medios obtenidos para las 50 ejecuciones de 24 horas del Recocido Simulado realizadas para cada operador. En gris se han resaltado los resultados del el operador mix-opt y sus variantes.

Operador	Valor objetivo medio (m)	Gap medio (respecto a la cota mínima)	Mejora del gap respecto al mejor operador de referencia (%) (Simple Inversion)
Mix-opt sin despl. de segmentos inter-veh.	1.928,23	0,028	62,07
Mix-opt sin intercambio unitario	1.928,67	0,028	61,75
Mix-opt sin desplazamiento unitario	1.929,44	0,029	61,2
Mix-opt sin operaciones unitarias	1.930,35	0,029	60,54
Mix-opt sin intercambios	1.931,12	0,030	59,99
Mix-opt	1.935,78	0,032	56,64
Mix-opt sin desplazamiento de segmentos	1.948,63	0,039	47,38
Mix-opt sin intercambio de segmentos	1.950,02	0,040	46,38
Mix-opt sin desplazamientos	1.995,03	0,064	13,98
Simple Inversion	2.014,46	0,074	0
LSM	2.052,23	0,094	-27,19*
Mix-opt sin operaciones con segmentos	2.076,43	0,107	-44,61*
Or-Opt	2.132,58	0,131	-77,35*
Relocate	2.139,99	0,141	-90,35*
Cut Inversion	2.146,70	0,145	-95,18*
Swap	2.666,56	0,422	-469,37*
Scramble	3.787,28	1,019	-1.276,05*

* Los porcentajes negativos indican peores resultados que los del operador referencia.

3.4.2. Estudio comparativo de los métodos de optimización

Al igual que con el operador mix-opt, una vez encontradas buenas configuraciones para cada uno de los métodos, se ejecutaron varias pruebas para determinar cuál de ellos era el mejor. Para ello se simuló un campo rectangular de dimensiones reales con 200 pistas de 3 m tal como se ilustra en la Figura 3.30. En este caso se construyó un campo de gran extensión considerando que las líneas de cultivo eran un 50% más largas que el ancho del campo, lo que arroja una superficie de 54 ha para el campo definido a tenor de los datos anteriores.

Los objetivos que se utilizaron para la comparación fueron el coste por insumos y el tiempo de realización de la tarea. El NSGA-II es capaz de optimizar ambos criterios a la vez mientras que en los casos del Recocido Simulado y del Algoritmo Genético se usó la ecuación (3.22) como objetivo para con ello optimizar ambos criterios de forma combinada. Los pesos establecidos para la combinación de costes fueron 1 y 27/3.600, de este modo se estimaba el coste de cada hora de trabajo de un operario a cargo de la supervisión de la flota en 27 €(Pedersen, Fountas, Have, & Blackmore, 2006).

Los costes de la flota son similares a los costes de referencia que se calcularon en la sección 3.2.3.1 (Tabla 3.3). En total se usaron 10 vehículos de 2,5 m de radio de giro para pulverizar localizadamente todo el campo. El número de dosis totales necesarias eran

288.000 y cada vehículo portaba un tanque con 20.000 dosis. El depósito para el reabastecimiento se supuso situado en la posición (300, 920), punto desde el que los vehículos debían también salir y volver al finalizar el plan de tratamiento asignado.

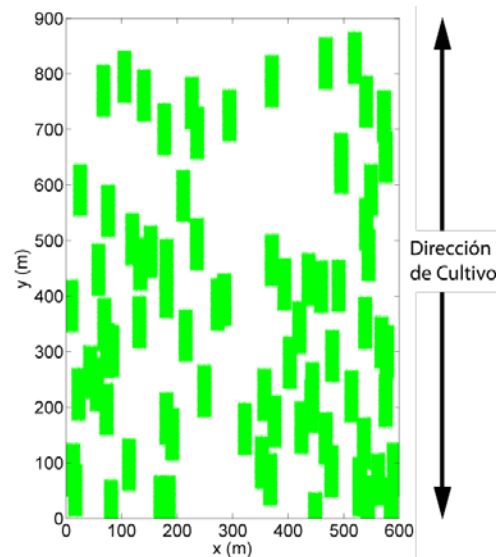


Figura 3.30. Campo de 54 ha con infestación de un 30 %. Mapa de distribución de mala hierba generado aleatoriamente

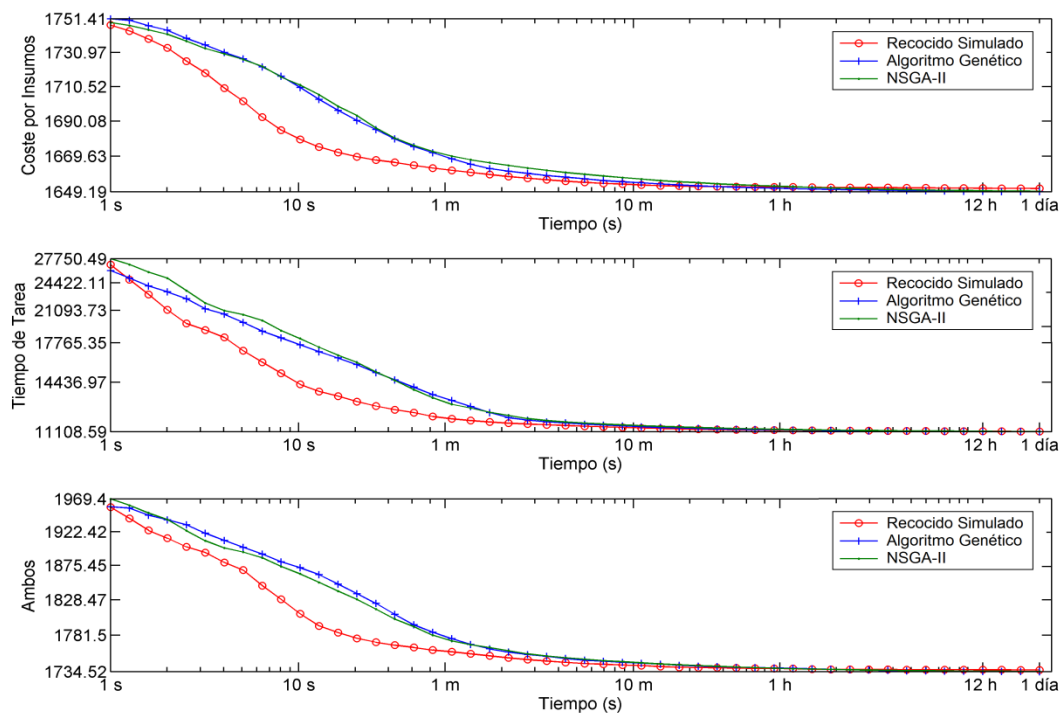


Figura 3.31. Convergencia media de cada algoritmo para cada criterio de optimización

Cada método se probó 50 veces durante 1 día entero y los resultados que se obtuvieron aparecen en la Figura 3.31 y en la Tabla 3.17. En la Figura 3.31 ambos ejes están en escala logarítmica para apreciar mejor la evolución de la mejor solución encontrada a lo largo del tiempo.

El Recocido Simulado y el Algoritmo Genético necesitaron ejecuciones distintas para optimizar cada criterio, mientras que el NSGA-II fue capaz de realizar las tres optimizaciones a la vez. Hasta la primera hora se puede afirmar que el mejor método es el Recocido Simulado, especialmente en el primer intervalo de 10 minutos, después las diferencias se hacen muy pequeñas y todos los resultados son muy parejos, aunque con el tiempo los métodos basados en población son capaces de obtener valores ligeramente mejores.

Tabla 3.17. Mejores resultados por método y criterio a lo largo del tiempo

Método Objetivo	Tiempo	Recocido Simulado	Algoritmo Genético	NSGA-II
Coste por Insumos (€)	10 s	1679,99*	1710,66	1711,85
	1 min	1661,89*	1669,21	1670,59
	10 min	1653,28*	1654,60	1656,85
	1 h	1651,97	1651,12*	1652,26
	12 h	1651,22	1649,42*	1649,74
	1 día	1651,09	1649,19*	1649,46
Tiempo (s)	10 s	14329,12*	17613,76	18330,36
	1 min	11941,31*	13238,13	12925,47
	10 min	11340,34*	11421,55	11485,35
	1 h	11195,17*	11216,53	11244,50
	12 h	11129,64	11120,15*	11129,89
	1 día	11124,29	11108,59*	11114,52
Ambos criterios combinados (€)	10 s	1811,42*	1872,69	1866,21
	1 min	1760,71*	1779,50	1776,23
	10 min	1742,47*	1745,60	1746,24
	1 h	1738,40*	1738,89	1738,60
	12 h	1736,87	1734,86*	1735,06
	1 día	1736,69	1734,52*	1734,65

*Mejor resultado hasta ese tiempo

En general los resultados para los tres métodos son muy parecidos, no obstante como el Recocido Simulado es el que converge más rápidamente, es preferible usar este método cuando exista alguna restricción en el tiempo de ejecución, en caso contrario es preferible utilizar el método NSGA-II ya que se obtienen resultados un poco mejores y además devuelve un conjunto de soluciones más diverso, donde las soluciones están optimizadas para cada uno de los criterios, de tal modo que es posible la elección de aquella solución que mejor encaja con la estrategia de manejo de la tarea agrícola a realizar en cada momento.

3.4.3. Pruebas ilustrativas

Después de mostrar la mejora en el rendimiento que supone la utilización del operador mix-opt propuesto y la mejora en la convergencia del Recocido Simulado en intervalos inferiores a 1 hora, se diseñaron diferentes test para comprobar, mediante simulación, la validez de un Planificador basado en un Recocido Simulado que integrase el operador mix-opt y, por ende, todo el planteamiento general detallado a lo largo del capítulo. Las pruebas en entornos reales con una flota de robots se reservaron para el siguiente capítulo, ya que en él se estudia en detalle el siguiente paso, en otras palabras la ejecución de la planificación y la supervisión que ésta conlleva.

Las pruebas establecidas tenían la finalidad de estudiar el rendimiento del Planificador desarrollado con escenarios teóricos de planificación basados en situaciones reales. Para todas las pruebas definidas las soluciones óptimas o al menos algunas de las características de éstas, eran fácilmente predecibles, por ejemplo, el número de vehículos a usar, la forma de las transiciones, las zonas a las que se dirigiría cada vehículo, etc., a partir de las características de la flota y el campo, por ello, se utilizaron como pruebas para comprobar que el Planificador general propuesto era capaz de obtener soluciones en consonancia con lo esperado. En la mayoría de estos test se consideró como tarea agrícola el control de malas hierbas a partir de la aplicación precisa de herbicida solo en las zonas afectadas.

Los campos que se consideraron en estas pruebas eran pequeños, de forma rectangular y estaban compuestos por 30 pistas de 3 m de anchura, que a su vez estaban formadas por calles de 0,75 m de ancho, la distancia habitual entre líneas de cultivo en maíz. En este estudio se consideró, además, que la barra de aplicación de herbicidas tenía 4 boquillas, cada una capaz de actuar sobre una calle. No se usaron campos con un mayor número de pistas para disponer de soluciones más o menos sencillas que fuesen fáciles de analizar.

Se utilizó una forma rectangular para todos los campos y una dirección de cultivo constante, es decir, todas las pistas eran rectas y tenían la misma longitud, lo que como se explicó con anterioridad no afecta a la complejidad del problema, ya que es el número de pistas más el número de vehículos lo que determina la complejidad del problema y no la forma, ni el ancho o el largo de las pistas.

La posición inicial de los vehículos en algunos problemas se fijó en el depósito (15,26) mientras que en otros casos se consideró que estaba en la parte superior de la primera pista o incluso no se fijó.

Por último, todas las pruebas se ejecutaron durante un periodo de 1 minuto, un tiempo que permite encontrar una buena solución para un problema no demasiado complejo como los definidos.

3.4.3.1. Test 1. Minimizar las distancias recorridas en cabecera por un único vehículo en función del radio de giro

El primer test se diseñó para evaluar si el Planificador era capaz de encontrar el orden de pistas óptimo para reducir la distancia recorrida, véase ecuación (3.16) y costes en metros en la Tabla 3.18. El punto de partida se situó en la parte superior de la primera pista y se empleó un único vehículo con varios radios de giro. El objetivo era recorrer todo el campo visitando una única vez cada pista. En este caso no se realiza aplicación de tratamiento, por lo que el plan generado por el Planificador estaba formado únicamente

por las trayectorias a seguir sin contener ninguna actuación asociada. El comportamiento del vehículo puede entenderse como el de una inspección o muestreo continuo.

Se probaron tres radios de giro: 1, 3 y 4 m. El objetivo era observar si, en todos los casos, las rutas obtenidas por el Planificador separaban las pistas de tal modo que las transiciones eran las más cortas posibles, es decir, de tipo Π , véase ecuación (2.1). La Figura 3.32 muestra las rutas obtenidas para cada caso.

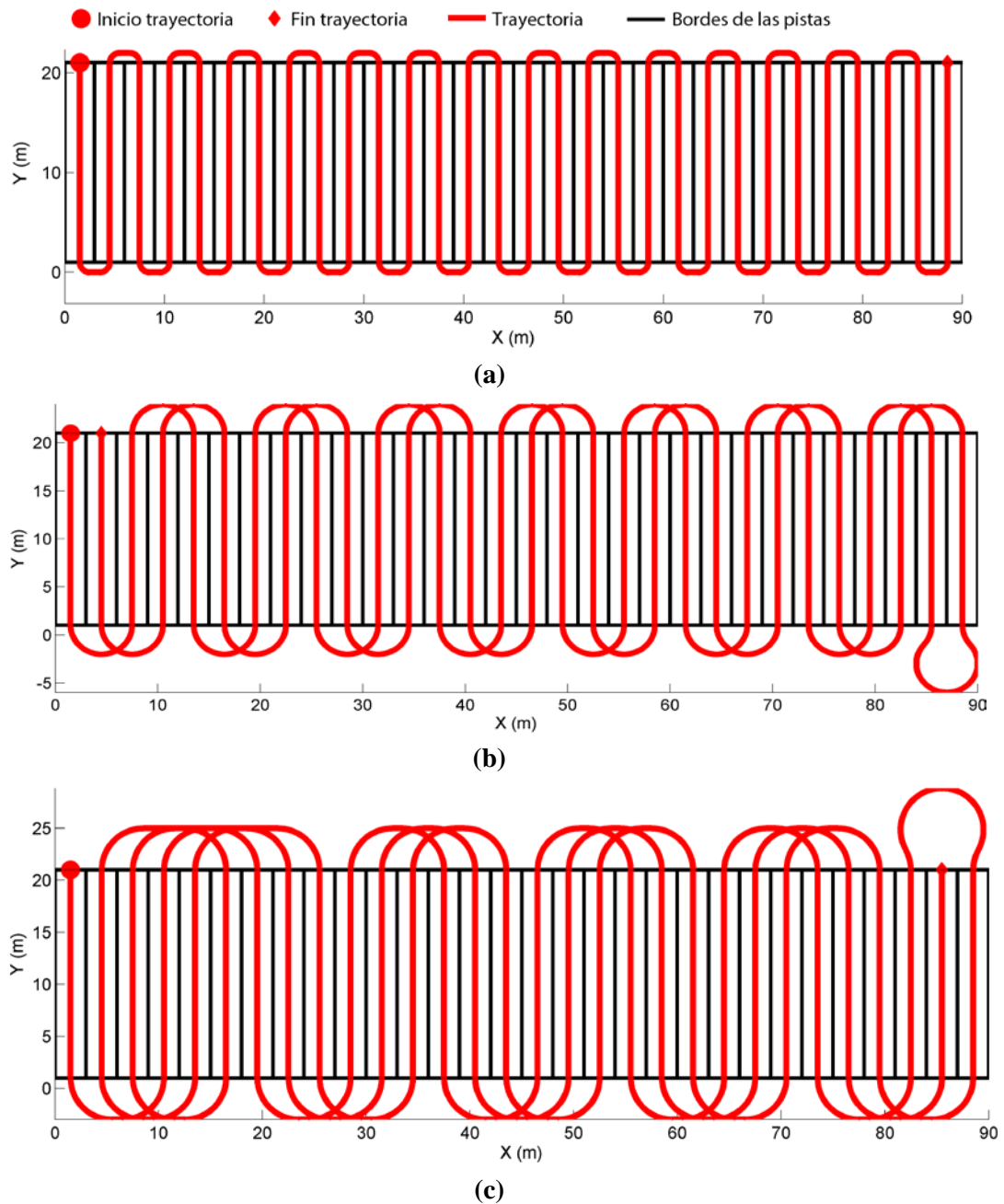


Figura 3.32. Rutas obtenidas en el test 1 usando un único vehículo y considerando distintos radios de giro: (a) 1 m, (b) 3 m y (c) 4 m.

Tabla 3.18. Costes usados en el test 1, ajustados para calcular solo la distancia en cabecera

Costes de cada operación (en metros)				
Arranque	Desp. inter-pista (1 m)	Desp. intra-pista (1 m)	Pulverización (1dosis)	Reabast.
0	1	0	0	0

Como se observa, cuanto mayor es el radio, más largos son los giros y las trayectorias. En los tres casos, las rutas obtenidas evitaron las transiciones entre pistas próximas cuando, por el radio de giro, solo era posible llevarlas a cabo mediante maniobras más costosas Ω , ecuación (2.2). En lugar de eso emplearon transiciones Π consiguiendo trayectorias más cortas aun cuando las transiciones eran entre pistas más separadas, véase ecuación (2.3).

En el primer caso (Figura 3.32a) el vehículo puede transitar perfectamente desde una pista a su adyacente implementando un giro de tipo Π , ya que el diámetro de giro de 2 m es inferior al ancho de la pista de 3 m, y por ello es capaz de recorrer el campo en orden, empezando por la primera pista y continuando siempre por la siguiente hasta alcanzar la última, siendo imposible cubrir el campo propuesto con un recorrido de menor distancia. En el segundo caso (Figura 3.32b), el giro más corto disponible de tipo Π implica un diámetro de 6 m, lo que obliga a dejar la pista adyacente sin cubrir y por tanto el vehículo salta una pista en casi todas las transiciones y tiene que volver para recorrer las pistas no visitadas en la primera pasada. La maniobra Ω que aparece se debe a que es imposible recorrer el campo por completo saltando siempre el mismo número de pistas. Cuando se llega a uno de los extremos de este modo, saltando siempre el mismo número de pistas, la única pista que está a la distancia apropiada es justo la última que se ha tratado y por tanto solo es posible continuar saltando hasta otra que esté más cerca o más lejos.

Análogamente, en el tercer caso (Figura 3.32c), el diámetro de giro es 8 m y la transición más corta de tipo Π requiere saltarse dos pistas como mínimo, que es lo que sucede en la mayoría de las transiciones de la solución hallada.

La Tabla 3.19 lista las soluciones encontradas y las distancias recorridas asociadas. Solo se muestra la parte vector permutación de las soluciones porque en este caso no se han considerado reabastecimientos.

Tabla 3.19. Distancias y soluciones del test 1 según el radio de giro considerado.

Radio (m)	Distancia (m)	Solución
		(Orden de las pistas)
1	120,11	(1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30)
3	281,99	(1,3,5,7,9,11,13,15,17,19,21,23,25,27,29,30,28,26,24,22,20,18,16,14,12,10,8,6,4,2)
4	412,51	(1,4,7,10,13,16,19,22,25,28,30,27,24,21,18,15,12,9,3,6,2,5,8,11,14,17,20,23,26,29)

3.4.3.2. Test 2. Minimizar las distancias recorridas en cabecera usando diferentes radios de giro y una flota de vehículos

El segundo test es similar al anterior, pero usando una flota de tres vehículos. Se diseñó con la finalidad de comprobar si el Planificador era capaz de minimizar aún más la distancia en caso de disponer de más de un vehículo, por ejemplo, distribuyendo los vehículos adecuadamente para evitar la necesidad de maniobras Ω .

Las posiciones iniciales de los tres vehículos se situaron en el extremo superior de las tres primeras pistas (véase Figura 3.33) y se consideraron los mismos radios de giro del test anterior, es decir, 1, 3 y 4 m.

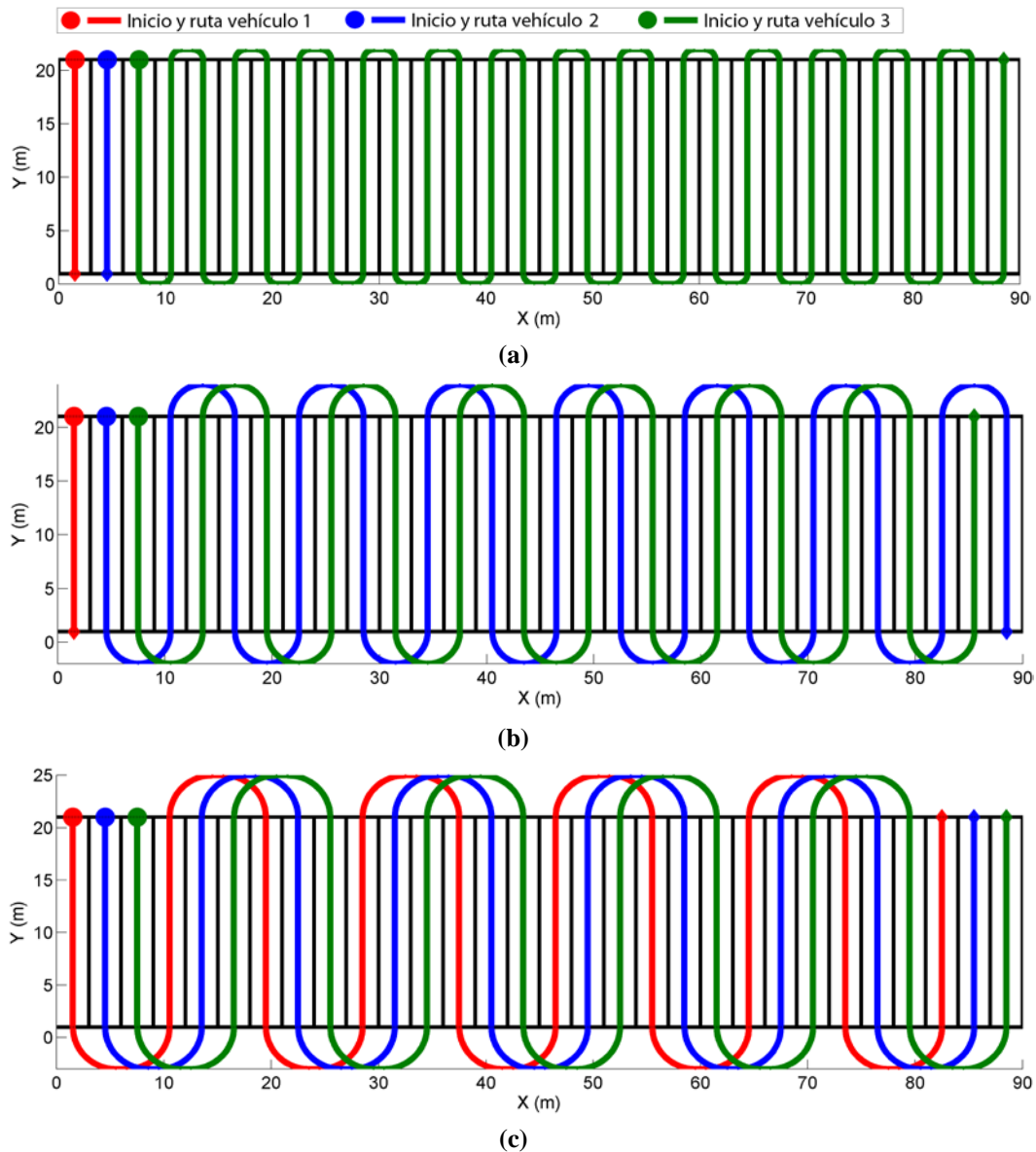


Figura 3.33. Rutas del test 2 usando varios vehículos y varios radios de giro: (a) 1 m, (b) 3 m y (c) 4 m.

En el primer caso (Figura 3.33a), como el radio de giro permitía transitar directamente de una pista a la siguiente usando maniobras Π , se usó el vehículo más a la derecha para recorrer el campo mientras que los otros dos vehículos solo cubrieron la pista que tenían más cerca. Esta solución tiene dos transiciones menos con respecto a la solución de la Figura 3.32a, de ahí que sea más corta.

En el segundo caso (Figura 3.33b), como solo era posible recorrer una mitad del campo aplicando transiciones mínimas Π , la solución proporcionada por el Planificador usó un segundo vehículo para completar la mitad intercalada que iba dejando el primer vehículo, de este modo se consigue que todas las transiciones sean mínimas. Además, el tercer vehículo cubre la pista más cercana ahorrando otra transición.

Finalmente, en el último caso (Figura 3.33c) cada vehículo solo puede cubrir un tercio del campo haciendo transiciones mínimas y por ello la solución involucra a los tres vehículos que actúan intercaladamente.

La Tabla 3.20 muestra las soluciones obtenidas. El símbolo “|” se utiliza para representar el elemento separador que distingue las pistas de cada vehículo. Como se puede observar, las distancias obtenidas son inferiores en todos los casos a las del test anterior.

Tabla 3.20. Distancias y soluciones del test 2 según el radio de giro, y comparadas con las distancias obtenidas en el test 1.

Resultados del test 2			Resultados del test 1
Radio (m)	Distancia (m)	Solución (Orden de las pistas)	Distancia (m)
1	111,82	(1 2 3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30)	120,11
3	254,47	(1 2,4,6,8,10,12,14,16,18,20,22,24,26,28,30 3,5,7,9,11,13,15,17,19,21,23,25,27,29)	281,99
4	366,29	(1,4,7,10,13,16,19,22,25,28 2,5,8,11,14,17,20,23,26,29 3,6,9,12,15,18,21,24,27,30)	412,51

3.4.3.3. Test 3. Minimizar la distancia recorrida en un tratamiento con una flota con tanques limitados

Este test se diseñó para comprobar si el Planificador era capaz de minimizar las distancias en casos en los que la capacidad del tanque influye en las rutas a seguir, como por ejemplo en el caso del tratamiento localizado con herbicida. Para ello se diseñó un campo de tal modo que solo se evitaba acudir al depósito para una distribución determinada de las pistas entre los vehículos. El objetivo era comprobar que el Planificador era capaz de encontrar esa distribución.

Se dispuso una flota con tres vehículos cuyos tanques podían almacenar respectivamente 10, 110 y 1110 dosis. A su vez, las 30 pistas del campo se dividieron de tal modo que 10 de ellas necesitaran solo 1 dosis de herbicida; otras 10, 11 dosis cada una y, por último, las 10 restantes necesitaban 110 dosis. Mediante esta distribución, la única forma de completar todo el campo sin reabastecimientos, consistía en usar el vehículo 1 para tratar las 10 pistas que requerían 1 dosis de tratamiento, el vehículo 2 para tratar las 10 pistas que requerían 11 dosis, y el vehículo 3 para tratar las restantes.

La Figura 3.34 muestra las rutas asociadas a los planes que se obtuvieron y, efectivamente, los vehículos se distribuyeron según lo esperado para evitar los reabastecimientos y minimizando la distancia recorrida. Se usó un radio de giro de 1 m para todos los vehículos de tal modo que el orden óptimo es el propio orden de las pistas

en el campo y en efecto esta es la mejor solución encontrada, es decir, (1,2,11,12,13,16,17,18,26,30|29,27,25,24,23,22,20,14,9,4|3,5,6,7,8,10,15,19,21,28).

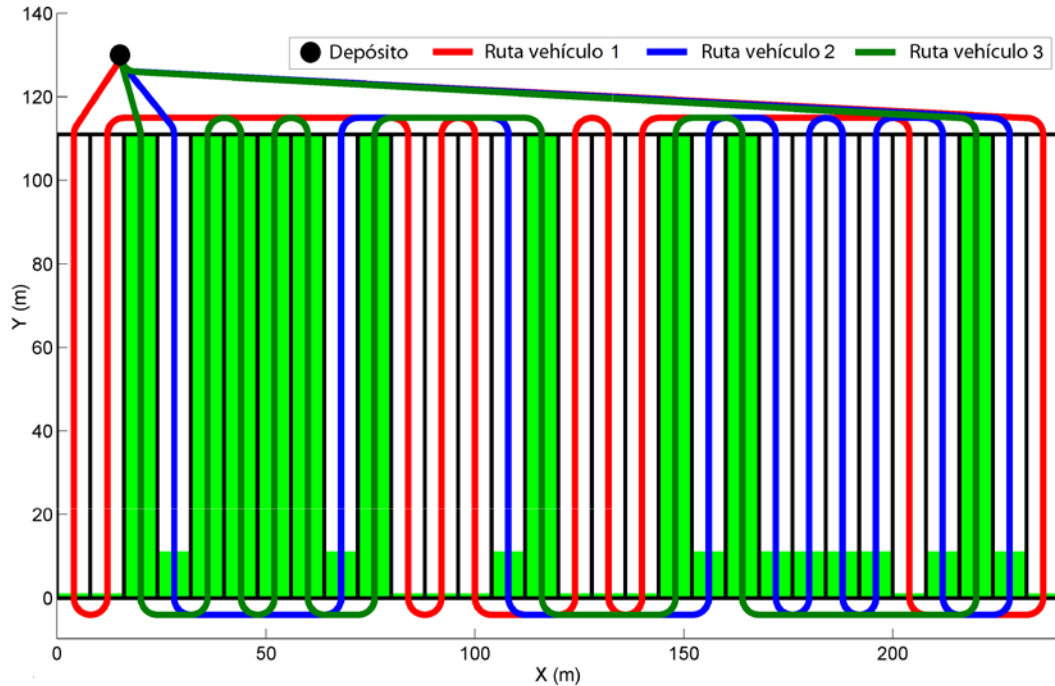


Figura 3.34. Rutas del test 3 usando una flota de 3 vehículos con tanques de 10, 110 y 1110 dosis y una distribución concreta de mala hierba.

3.4.3.4. Test 4. Minimizar el coste por insumos, el tiempo de tarea y ambos criterios a la vez abordando el tratamiento con una flota de vehículos homogéneos

Este test se diseñó para comprobar cómo las rutas óptimas asociadas a los planes de tratamiento pueden variar dependiendo del criterio de optimización y mostrar como el Planificador era capaz de encontrar esas soluciones óptimas.

En este caso se usaron 5 vehículos homogéneos, es decir, con costes y capacidades iguales (véase Tabla 3.21) para pulverizar localizadamente un campo con malas hierbas. Como criterios de optimización se estudiaron los siguientes: (a) el coste por insumos utilizados, (b) el tiempo requerido para la realización de la tarea y (c) ambos criterios a la vez mediante una combinación lineal según la ecuación (3.22) con los pesos 1 y 27/3600.

Los costes se obtuvieron a partir de los consumos de referencia comentados en la sección 3.2.3.1. El campo se dividió en 2.400 celdas de 75 x 75 cm y se supuso una infestación del 30 %, por lo que 720 de las celdas elegidas aleatoriamente estaban infestadas. La capacidad de almacenamiento de los vehículos se fijó en 720 dosis, para que fuera posible tratar todo el campo con un único vehículo sin necesidad de reabastecimiento. Para todos los vehículos se consideró el mismo punto de arranque, el punto (15, 26). La Figura 3.35 muestra las rutas asociadas a los planes que se obtuvieron para cada criterio.

Tabla 3.21. Características de los vehículos usados en el test 4

Costes por operación										Radio de giro (m)	Capacidad de tanque (dosis de herbicida)
Arranque		Desp. inter-pistas (1 m)(10 km/h)		Desp. intra-pistas (1 m)(6 km/h)		Pulverización (1 dosis)		Reabast.			
€	t(s)	€(10 ⁻³)	t (s)	€(10 ⁻³)	t (s)	€(10 ⁻²)	t (s)	€	t (s)		
0,2	0	0,96	0,36	0,96	0,6	0,51	0	0,0	120	3	720

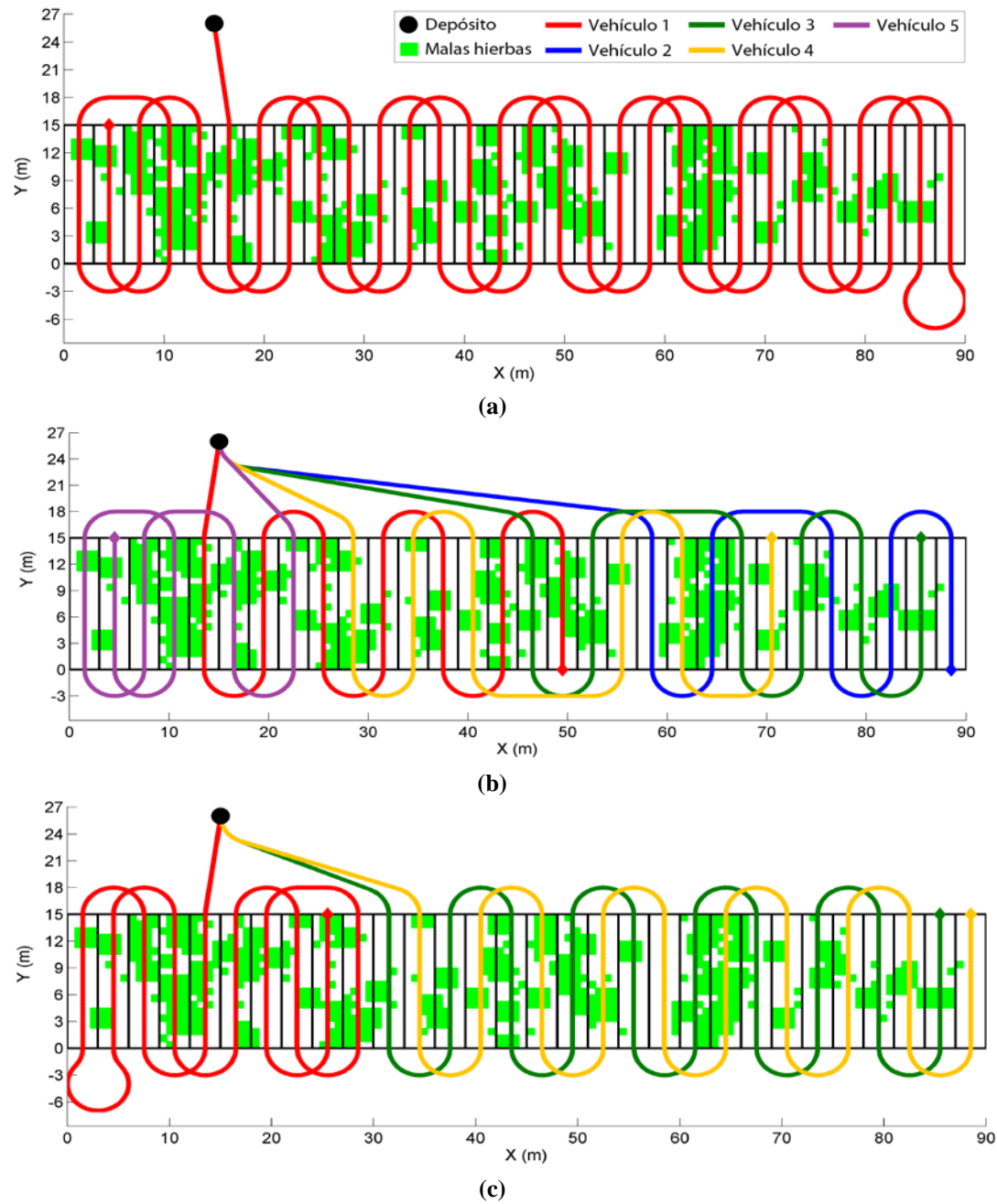


Figura 3.35. Rutas obtenidas en el test 4 para una flota de robot con iguales características y optimizando distintos criterios: (a) coste por insumos, (b) tiempo de tarea y (c) ambos.

Al igual que ocurrió en los ensayos anteriores, en todas las rutas obtenidas se evitaron las maniobras Ω al máximo. Las dos únicas maniobras Ω que aparecieron eran necesarias para que todas las demás maniobras fueran Π y mínimas.

Para el coste por insumos, la mejor solución solo usa un vehículo porque existe una penalización por arranque (0,2 €) por cada unidad extra que se utiliza y porque un único vehículo tiene un tanque suficiente para completar toda la tarea sin necesidad de reabastecimiento.

Puesto que, para este test, las soluciones siempre van a gastar la misma cantidad por herbicida (todos los vehículos tienen el mismo coste de pulverización y siempre se van a aplicar el mismo número de dosis), el Planificador focalizó la optimización en minimizarla distancia recorrida para reducir el gasto en combustible. Como se puede observar la solución en este caso es distinta a la que se obtuvo en el primer test (Figura 3.32b) partiendo de un campo igual y un mismo radio de giro (3 m), la razón es que la posición inicial considerada no es la misma. En el presente test se parte del depósito y la distancia sería un poco superior si se utilizasen las trayectorias obtenidas en el test 1b. En concreto con la ruta hallada la distancia recorrida es de 296,096 m mientras que si se hubiera utilizado la solución del test 1b pero partiendo desde el depósito la distancia hubiera sido de 300,272 m.

En todas las repeticiones que se realizaron de esta prueba, la solución siempre fue la misma diferenciándose unas de otras sólo en el vehículo involucrado en cada caso, algo obvio si se tiene en cuenta que todos los vehículos de la flota son iguales.

Cuando el criterio a optimizar es el tiempo la solución involucra a todos los vehículos de la flota (véase Figura 3.35b), ya que es la forma de paralelizar la ejecución de la tarea al máximo. Es más, como los vehículos son indistinguibles la solución asigna un número parecido de pistas a cada vehículo, en concreto las 30 pistas se distribuyen como sigue: 7, 5, 6, 6 y 6. El número asignado a cada vehículo no es exactamente el mismo, porque el punto de partida está en la parte izquierda del campo, por lo que, por ejemplo, el vehículo 2, que es el que más se aleja para tratar las pistas más a la derecha del campo, compensa el mayor tiempo de desplazamiento con la asignación de un menor número de pistas que tratar. Análogamente sucede con el vehículo 1, que tiene un trayecto muy corto desde el depósito hasta su primera pista y por tanto se puede hacer cargo de un mayor número de pistas.

Por último, la solución encontrada para minimizar ambos criterios simultáneamente (Figura 3.35c) está a medio camino entre las dos soluciones anteriores. Más concretamente usa tres vehículos porque el tiempo convertido a euros penaliza mucho el coste total y, simplemente usando dos unidades más, en teoría el trabajo debería realizarse hasta un 67 % más rápido. En la práctica se reduce incluso más porque también se ahorra distancia extra en transiciones. Tal como se ha diseñado el test, una cuarta unidad o incluso una quinta no compensaban debido a los costes por arranque.

La Tabla 3.22 muestra los costes de las soluciones para el criterio para el que fueron optimizadas y también los costos de las mismas soluciones para los otros criterios. De este modo es fácil observar que efectivamente las soluciones obtuvieron siempre los mejores costes para los criterios para los que fueron optimizados pero no así para los otros criterios. Es interesante observar que los valores de la columna “ambos” pueden calcularse a partir de los otros dos costes usando la ecuación (3.22), que es la que se utilizó para optimizar, y los coeficientes 1 y 27/3600.

Los costes totales son muy bajos porque el campo es pequeño, concretamente 1.350 m²; con un campo mayor los costes subirían bastante. Por ejemplo, en caso de que

el campo hubiera sido de 4 ha, un tamaño que ya empieza a ser productivo, los precios finales hubieran estado respectivamente alrededor de 124 €, 28 minutos y 137 € para el coste por insumos, el tiempo y ambos a la vez. Para un campo comercial de 20 ha, tamaño frecuente en campos comerciales de cultivo extensivo, estaríamos hablando de 607 €, 2 horas y 17 minutos, y 669 €, respectivamente. Por último, para campos considerados muy grandes, de alrededor de 100 ha, las cantidades estarían en 3.100 €, 11 horas y 40 minutos, y 3.415 €, respectivamente.

Tabla 3.22. Coste de cada solución obtenida para cada criterio de optimización en el test 4. Solo se muestra el vector de pistas de las soluciones porque no fue necesario ningún reabastecimiento para ninguno de los tres casos.

Soluciones optimizadas para	Costes	Coste por Insumos (€)	Tiempo (s)	Ambos (€)
Coste por insumos	(6,8,10,12,14,16,18,20,22,24,26,28,30,29,27,25,23,21,19,17,15,13,11,9,7,5,3,1,4,2, , , ,)	4,59*	376,59	7,41
Tiempo	(5,7,9,11,13,15,17 20,22,26,28,30 16,18,23,25,27,29 10,12,14,19,21,24 8,6,3,1,4,2)	5,48	87,38*	6,14
Ambos	(5,3,1,2,4,6,8,10,7,9, , ,11,13,15,17,19,21,23,25,27,29 12,14,16,18,20,22,24,26,28,30)	5,01	129,11	5,98*

*Coste de la solución óptima para cada criterio

3.4.3.5. Test 5. Minimizar el coste por insumos, el tiempo de tarea y ambos a la vez abordando el tratamiento con una flota de vehículos homogéneos con tanques pequeños

En los test anteriores los vehículos tenían un tanque con capacidad suficiente para tratar el campo entero sin paradas de reabastecimiento. Si los tanques hubieran sido menores, los vehículos se hubieran visto probablemente obligados a acudir al depósito para reponer herbicida o a colaborar y sumar sus tanques realizando la tarea entre varios. Para comprobar que el Planificador genera este tipo de soluciones, se diseñó el test 5, similar al anterior salvo en que ahora la capacidad de los tanques es 144 dosis, es decir, lo necesario para tratar exactamente un quinto de los rodales de malas hierbas del campo.

La Figura 3.36 muestra las rutas que se obtuvieron. Se puede observar que efectivamente la solución calculada para minimizar el coste por insumos (Figura 3.36a) utiliza un único vehículo y son además necesarios 5 reabastecimientos, en concreto al terminar el tratamiento de las pistas 1, 2, 7, 15 y 17. Estos reabastecimientos no son necesarios en el test 4a tal como se puede ver en la Figura 3.35a. Aunque en un principio lo esperable era una solución con 4 reabastecimientos en lugar de 5, el reabastecimiento extra se debe a que las soluciones con un repostaje menos realizan siempre algunas de las transiciones hacia el depósito desde pistas muy alejadas, lo que implica un mayor desplazamiento y gasto en combustible; de ahí que compense hacer un repostaje más porque así la distancia total recorrida es más corta. Asimismo, la limitada capacidad del tanque también ha forzado un orden de pistas diferente al de la Figura 3.35a para que el depósito no quede muy lejos de las transiciones con reabastecimientos.

Una de las hipótesis que en un principio parecía razonable, era suponer que el Planificador encontraría una solución que usara varios vehículos, para así, entre los tanques de todos, evitar los costes por reabastecimiento, sin embargo, en este caso

concreto no fue así debido a los costes que se han considerado de partida, ya que el coste por arranque, 0,2 € era superior al coste por reabastecimiento, 0 € más el coste del combustible necesario para ir y volver del depósito.

En cuanto a la optimización del segundo criterio (tiempo), la ruta encontrada (Figura 3.36b) presenta características similares a la solución obtenida para el test 4b, ya que en este caso también se distribuyó el trabajo de modo equitativo entre la flota. En concreto los vehículos del 1 al 5 tratan respectivamente 5, 7, 6, 6 y 6 pistas, aunque el orden de pistas es diferente al obtenido en el test anterior. En el test 4b los vehículos aplicaban respectivamente 193, 98, 116, 145 y 168 dosis, reparto que no es válido bajo las condiciones del test 5b por el nuevo tamaño del tanque, 144 dosis. En la nueva solución todos los vehículos aplican las mismas dosis que coincide con la máxima cantidad de almacenamiento, 144, y así, a pesar de lo ajustado de la situación, se consiguen evitar los reabastecimientos.

Por último, cuando el objetivo es optimizar los dos criterios a la vez, la solución encontrada involucra a todos los vehículos de la flota para evitar las penalizaciones por reabastecimiento, a pesar de incurrir en un coste mayor por el arranque de los vehículos. Esto se debe a que la penalización por tiempo de repostaje es de 120 s que convertidos a euros suponen 0,9 € –véase ecuación (3.22) con los pesos 1 y 27/3600–, por tanto una penalización bastante mayor que los 0,2 € que cuesta el arranque de un vehículo. En efecto, el impacto del tiempo en el coste combinado de la solución tiene tanto peso, que la solución para ambos criterios es similar a la obtenida cuando se optimiza el tiempo, salvo por el ya explicado baile de vehículos debido a que estos tienen las mismas características y por tanto son intercambiables.

Al igual que en los casos anteriores, las pruebas se repitieron varias veces y los resultados fueron siempre los mismos.

La Tabla 3.23 muestra los costes de las soluciones encontradas y se puede observar que estos son superiores a los que aparecen en la Tabla 3.22 debido a los costes de reabastecimiento y las distancias extra, producto de las limitaciones impuestas en la capacidad del tanque.

Se realizó un último test para comprobar si un coste por reabastecimiento superior podría llegar a compensar el coste por arranque, en otras palabras, si subiendo el coste de reabastecer, el Planificador generaba una solución que usase más vehículos para el test 5a. Así, se aumentó el coste por repostaje a 0,2 € es decir, el repostaje tenía el mismo coste que el arranque de un vehículo. En estas condiciones el Planificador generó soluciones que involucraban a los 5 vehículos (véase Figura 3.37) y con un coste de 5,73 € y por tanto inferior a los 6,04 € de coste que tenía la solución con un solo vehículo y cinco repostajes. Es importante resaltar que, a pesar de que los costes de partida son iguales, la solución con repostajes es más costosa porque el vehículo tiene que recorrer una distancia extra cada vez que realiza un repostaje (ida y vuelta al depósito) y por tanto consume algo más de combustible. Aun así, en la solución aparece un reabastecimiento en una transición cercana al depósito (entre la pista 6 y la 4) para que la flota no vaya tan ajustada, es decir, disponga de más dosis y puedan aparecer transiciones más cortas que acaben beneficiando al coste global.

La secuencia de pistas para esta nueva solución es (3,5,2,6,4,1|13,15,22,26,24|10,12,21,27,29,30|7,9,14,16,18|8,11,19,17,20,23,25,28).

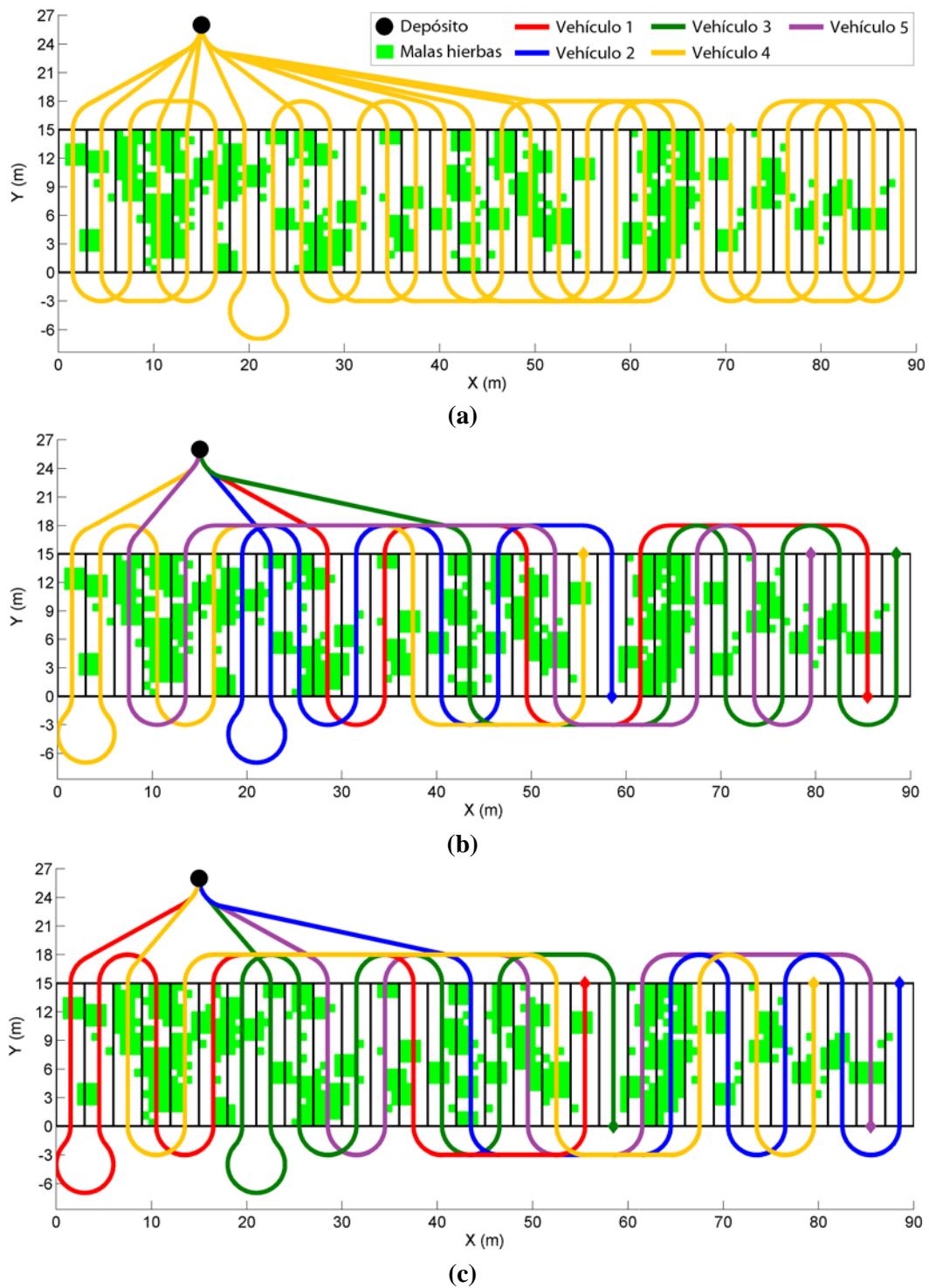


Figura 3.36. Rutas obtenidas en el test 5 para una flota de robots con iguales características y optimizando distintos criterios: (a) coste por insumos, (b) tiempo requerido para la realización de la tarea y (c) los criterios anteriores simultáneamente

capaz de repartir el trabajo aprovechando convenientemente las nuevas características de la flota. La Figura 3.38 muestra las soluciones generadas.

Tabla 3.24. Costes en tiempo de las operaciones de desplazamiento usadas en el test 6

Vehículo	Velocidad (km/h)	Desplazamiento Inter-pista (1 m)	Desplazamiento Intra-pista (1 m)
1	12	0,30 s	0,30 s
2	10	0,36 s	0,36 s
3	8	0,45 s	0,45 s
4	6	0,60 s	0,60 s
5	4	0,90 s	0,90 s

Como la flota sigue siendo homogénea en términos del coste por insumos, la ruta que minimiza este coste (véase Figura 3.38a) es de nuevo la misma que se encontró en el test 4a, como cabía esperar, en otras palabras, aquella que minimiza la distancia, salvo que en este caso se emplea el segundo vehículo. Sin embargo, cuando el criterio que se quiere minimizar es el tiempo, sí aparecen diferencias con respecto a la solución hallada anteriormente. En esta ocasión los vehículos no trabajan con el mismo número de pistas, sino que hay un reparto proporcional a la velocidad del vehículo, de forma que cuanto más rápido es el vehículo más pistas trata, en concreto, los vehículos de la flota considerada tratan 9, 7, 6, 5 y 3 pistas respectivamente. De este modo se consigue que todos los vehículos tarden más o menos el mismo tiempo en realizar la tarea asignada y el tiempo total no se dispare porque alguno de ellos continúe trabajando mientras los demás ya han acabado.

Finalmente, para la optimización simultánea de los dos costes, se halló, como en casos anteriores, una solución intermedia con un número intermedio de vehículos, en concreto los tres más rápidos ya que el coste asociado al tiempo de ejecución tiene un gran impacto en el coste combinado. La Tabla 3.25 lista los costes obtenidos para cada solución y cada criterio optimizado.

Tabla 3.25. Costes de cada solución obtenida para cada criterio de optimización en el test 6. Solo se muestra el vector de pistas de las soluciones porque no fue necesario ningún reabastecimiento para ninguno de los tres casos.

Soluciones optimizadas para	Costes	Coste Insumos (€)	Tiempo (s)	Ambos (€)
Coste insumos	(0,6,8,10,12,14,16,18,20,22,24,26,28,30 29,27,25,23,21,19,17,15,13,11,9,7,5,3,1,4,2 , ,)	4,59*	268,59	6,60
Tiempo	(13,16,18,20,22,24,26,28,30 17,19,21,23,25,27,29 2,4,7,9,11,15 6,8,10,12,14 5,3,1)	5,44	74,30*	6,00
Ambos	(8,10,12,14,16,18,20,22,24,26,28,30 11,13,15,17,19,21,23,25,27,29 6,9,7,5,3,1,4,2 , ,)	5,00	92,10	5,69*

*Coste de la solución óptima para el criterio

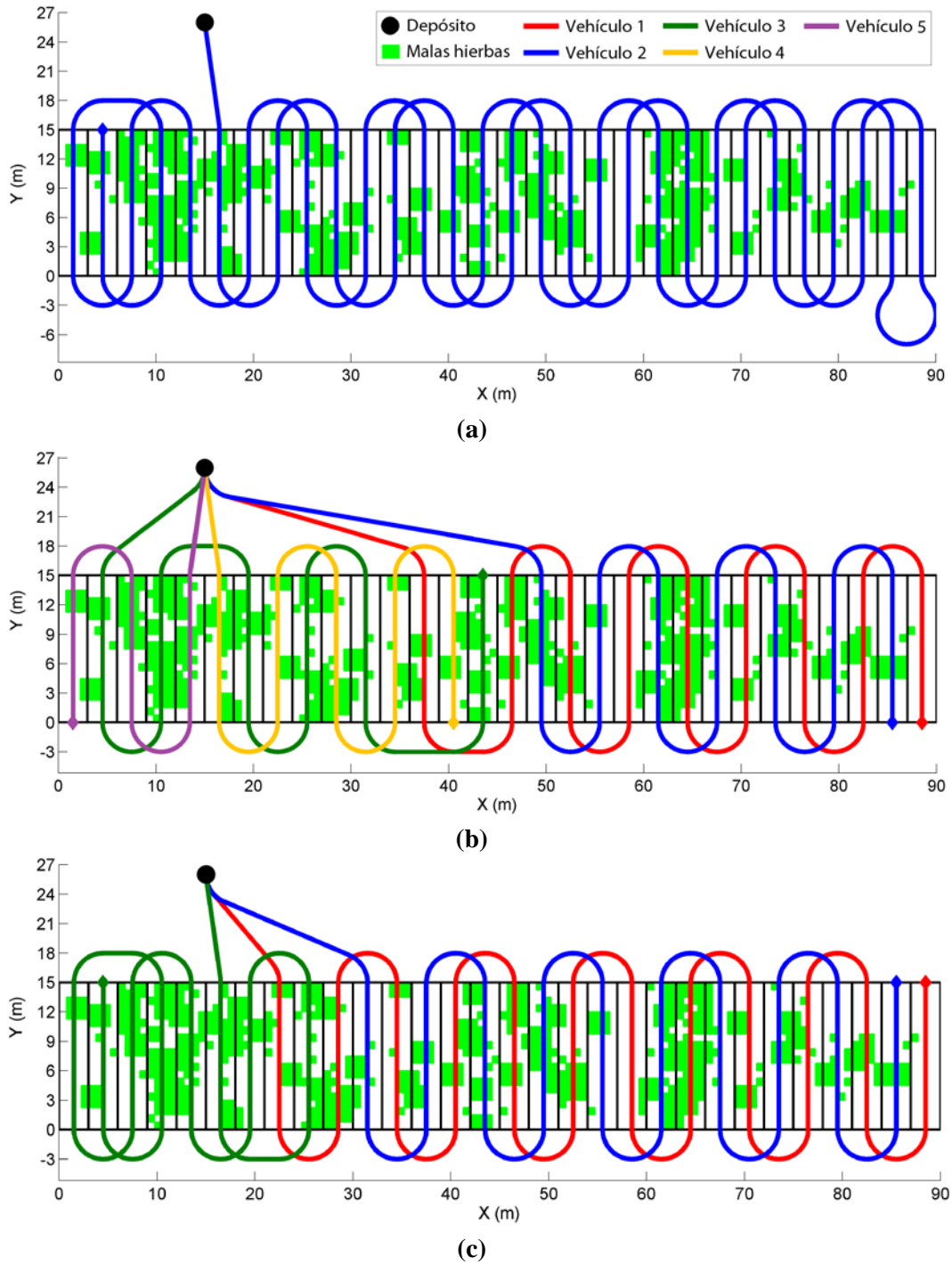


Figura 3.38. Rutas obtenidas en el test 6 para una flota de robots con distintas velocidades y optimizando distintos criterios: (a) coste por insumos, (b) tiempo requerido para la realización de la tarea y (c) los criterios anteriores simultáneamente.

Para comprobar si el Planificador era también capaz de dar prioridad a los mejores vehículos en el caso del coste por insumo, se variaron levemente los costes por combustible como se indica en la Tabla 3.26, suponiendo diferentes consumos debidos, por ejemplo, a la antigüedad del vehículo. Con estos costes, la ruta asociada al plan que encontró el Planificador fue la misma que para los ensayos 4a y 6a, es decir, aquella que minimiza la distancia, aunque en este caso el Planificador siempre genera soluciones en las que aparece el primer vehículo porque es el que menos consume.

Tabla 3.26. Costes por desplazamiento según vehículo

Vehículo	Desplazamiento Inter-pista (€/m)	Desplazamiento Intra-pista (€/m)
1	0,00096	0,00096
2	0,00097	0,00097
3	0,00098	0,00098
4	0,00099	0,00099
5	0,00100	0,00100

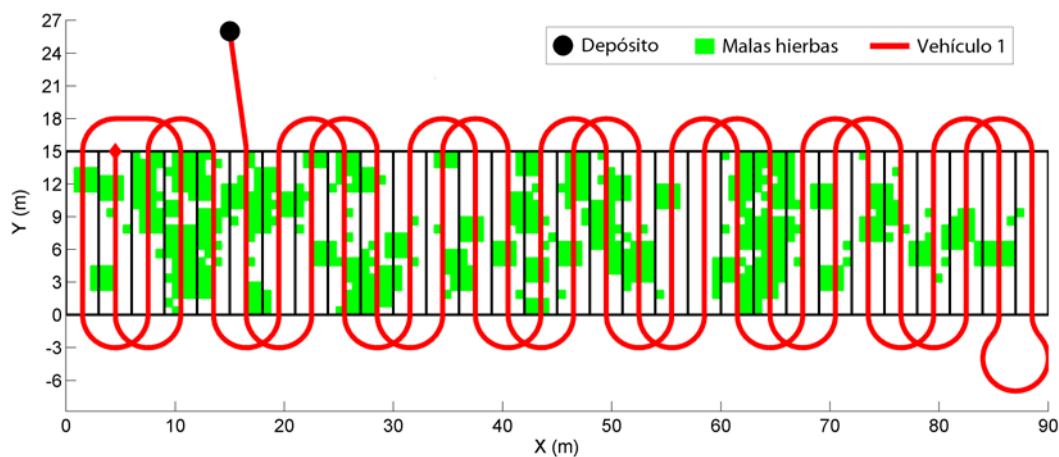


Figura 3.39. Ruta obtenida que minimiza el coste por insumos cuando los costes por desplazamiento son los que se muestran en la Tabla 3.26

3.4.3.7. Test 7. Minimizar el coste por insumos abordando el tratamiento con una flota de vehículos heterogéneos con costes contrapuestos

Como se comprobó en el test anterior, en general el Planificador usa preferentemente los vehículos con los costes más pequeños para llevar a cabo la tarea. Sin embargo, puede suceder que los vehículos tengan costes muy bajos para algunas operaciones y muy altos para otras. Por ejemplo, puede pasar que unos vehículos sean muy buenos pulverizando, por tener aperos muy precisos que no desperdicien nada de herbicida y, sin embargo, sean malos para desplazarse, por tener un consumo muy alto de combustible (por ejemplo, por la cilindrada del motor o un peor mantenimiento a lo largo de los años), y viceversa, puede haber vehículos con consumo pequeño de combustible pero alto de herbicida. En estos casos no está tan claro qué vehículos usar y cuándo

hacerlo. Por ejemplo, puede suceder que sea rentable emplear los vehículos con un coste bajo en combustible y alto en herbicida en aquellas zonas con poca cantidad de malas hierbas, mientras que los vehículos con un coste alto en combustible y bajo en herbicida sean más apropiados para las zonas más infestadas. Para comprobar si el Planificador era capaz de encontrar buenas soluciones en estos casos, se diseñó este test con una flota con costes contrapuestos, véase Tabla 3.27, calculados para producir este tipo de situaciones en un campo con una longitud de pista similar a las pruebas realizadas hasta el momento, es decir, 15 m.

Tabla 3.27. Costes considerados por insumos para los vehículos de la flota en el test 7

Vehículo	Costes por operación (€)				
	Arranque	Desplazamiento Inter-pista (1 m)	Desplazamiento Intra-pista (1 m)	Pulverización (1 dosis)	Reabast.
1	0	0	0,0008	0,005400	>0
2	0	0	0,0009	0,005306	>0
3	0	0	0,0010	0,005259	>0
4	0	0	0,0011	0,005228	>0
5	0	0	0,0012	0,005205	>0

A partir de la longitud de las pistas y del número de malas hierbas que contienen, es posible calcular el coste por insumos de cada pista aplicando la ecuación (3.30):

$$coste(p, v) = c_{o_d, v}^I \cdot longitud(p) + c_{o_p, v}^I \cdot dosis(p) \quad (3.30)$$

donde $c_{o_d, v}^I$ y $c_{o_p, v}^I$ son los costes en insumos por desplazamientos intra-pista y pulverización del vehículo v (listados en la Tabla 3.27), mientras que $longitud(p)$ y $dosis(p)$ son respectivamente la longitud en metros de la pista p y el número de dosis necesarias para tratarla que coincide con el número de celdas afectadas por malas hierbas.

Aplicando la ecuación (3.30), la Tabla 3.28 muestra los posibles costes de tratar una pista según el número de malas hierbas que contenga y el vehículo que se emplee.

Tabla 3.28. Coste por insumos de tratar una pista según el número de malas hierbas que contenga y el vehículo que se emplee

Número de malas hierbas	Coste por insumos de tratar una pista (€)				
	Usando vehículo 1	Usando vehículo 2	Usando vehículo 3	Usando vehículo 4	Usando vehículo 5
0-16	0,01200-0,09840*	0,01350-0,09840	0,01500-0,09915	0,01650-0,10015	0,01800-0,10128
17-32	0,10380-0,18480	0,10371-0,18330*	0,10441-0,18330	0,10538-0,18380	0,10648-0,18455
33-48	0,19020-0,27120	0,18861-0,26820	0,18856-0,26745*	0,18903-0,26745	0,18975-0,26783
49-64	0,27660-0,35760	0,27351-0,35310	0,27271-0,35160	0,27268-0,35110*	0,27303-0,35110
65-80	0,36300-0,44400	0,35840-0,43800	0,35686-0,43575	0,35633-0,43475	0,35630-0,43438*

*Tramo con coste más bajo

A partir de la Tabla 3.27 se puede afirmar que el vehículo 1 es el óptimo para las pistas que contengan de 0 a 16 celdas afectadas por malas hierbas; el vehículo 2, para las pistas con valores comprendidos entre 17 y 32; el vehículo 3 cuando están afectadas de 33 a 48 celdas; el vehículo 4 para las de 49 a 64 celdas, y, por último, el vehículo 5 para las de 65 a 80. El campo de la Figura 3.40 se diseñó con un número creciente de malas hierbas por pista para comprobar si la solución del Planificador distribuía los vehículos acorde a lo indicado en la Tabla 3.28. No se condicionó la posición inicial ni se contabilizaron los costes de los desplazamientos inter-pista para no modificar la distribución esperada. Al ser 0 el coste de los desplazamientos inter-pista, el Planificador no evitó la realización de giros Ω , ya que el orden para conectar las pistas no era el objeto de este test. El coste por reabastecimiento se estableció a un valor mayor que cero para evitar que aparecieran repostajes; nótese que si el coste de reabastecimiento se hubiera puesto a 0, el Planificador no habría evitado la generación de soluciones que los incluyesen.

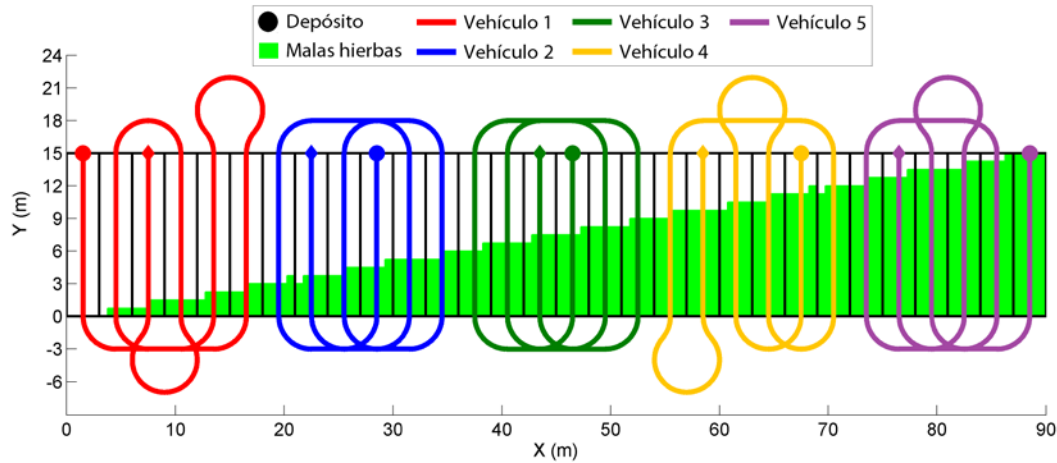


Figura 3.40. Rutas obtenidas para el test 7 usando costes contrapuestos para el coste de desplazarse y el coste de pulverizar

3.4.3.8. Test 8. Minimizar el tiempo abordando el tratamiento con una flota de vehículos homogéneos en un campo compuesto por varias parcelas

Este test se diseñó para mostrar que el Planificador también puede trabajar con campos formados por varias parcelas. El problema sigue siendo el mismo, es decir, encontrar el mejor orden en el que recorrer las pistas para que un determinado criterio sea óptimo. La única diferencia radica en que ahora no todas las pistas se van sucediendo adyacentemente, sino que pueden estar separadas por terreno que no es cultivo cuando pertenecen a diferentes parcelas. En el test se usó una flota con 3 vehículos homogéneos que partían desde el depósito y el objetivo era minimizar el tiempo. Como las parcelas son de tamaño parecido y el depósito estaba en una posición intermedia, el Planificador asignó una parcela a cada vehículo (véase Figura 3.41). En este caso se utilizó un radio de giro pequeño para todos los vehículos por claridad, para que no aparecieran transiciones

más largas que pudieran descompensar el tiempo empleado por cada vehículo y que produjeran un reparto menos claro.

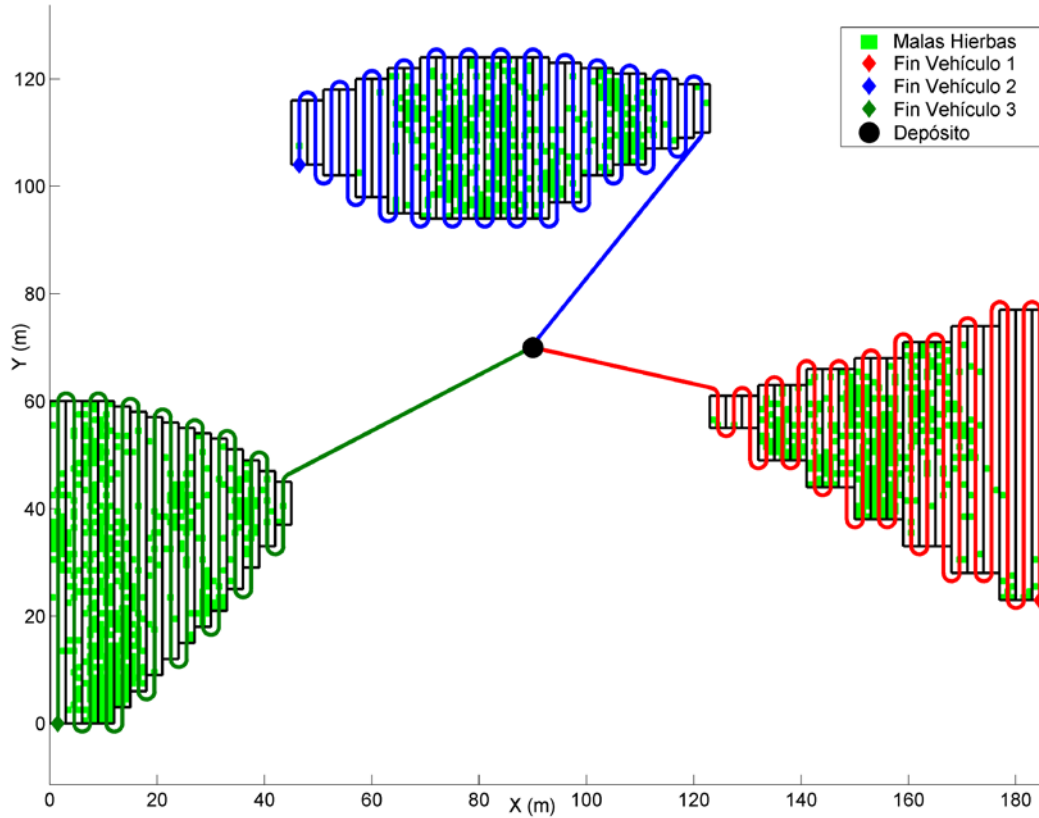


Figura 3.41. Rutas obtenidas para el test 8 usando una flota de 3 vehículos homogéneos y minimizando el tiempo de tarea.

3.4.4. Comparación con otros autores

Una vez comprobado que el Planificador con el planteamiento general propuesto es capaz de obtener planificaciones razonables en una amplia variedad de situaciones, se llevaron a cabo algunas pruebas más para comparar los resultados del Planificador propuesto con los resultados de algunas de las aproximaciones que se encuentran en la bibliografía más relevante. En concreto con los trabajos de (Bochtis & Vougioukas, 2008) y de (Hameed, Bochtis, & Sorensen, 2011), que además de ser de gran relevancia, proporcionan información suficiente sobre los escenarios utilizados lo que permite una reconstrucción fiel de los problemas con los que probar el Planificador desarrollado. Para la comparación con la aproximación propuesta en (Bochtis & Vougioukas, 2008) se configuró el Planificador para que generase soluciones para un único vehículo y se utilizó la distancia recorrida como criterio a optimizar. En definitiva el objetivo era encontrar las rutas que recorrieran en una distancia mínima 4 campos. En el primer caso, el campo (campo1) estaba formado por 8 pistas de 2,89 m de ancho y 30 de longitud, mientras que el radio de giro del tractor era 3,5 m. La solución obtenida por el Planificador propuesto

(véanse resultados en las Tabla 3.29 y Tabla 3.30) ordenaba el recorrido de las pistas del siguiente modo: (1,4,7,3,6,2,5,8), con una distancia total asociada de 334,439 m, de los cuales 94,439 m eran metros recorridos en cabecera. Para el mismo problema los otros autores obtenían la secuencia de pistas: (1,4,7,3,6,8,5,2), que daba lugar a una distancia recorrida de 335,767 m, siendo 95,767 m en cabecera, es decir, un resultado ligeramente peor. Cabe señalar que ambas distancias se calcularon a partir de la secuencia de pistas propuesta por cada Planificador utilizando el mismo procedimiento junto con las ecuaciones (3.16) y (3.19). Es importante recordar que el margen de mejora está en la distancia recorrida en cabecera ya que la distancia dentro de las pistas no se puede reducir. La Figura 3.42 muestra de izquierda a derecha la ruta de la solución generada por la aproximación propuesta por (Bochtis & Vougioukas, 2008) y la ruta de la solución proporcionada por Planificador desarrollado

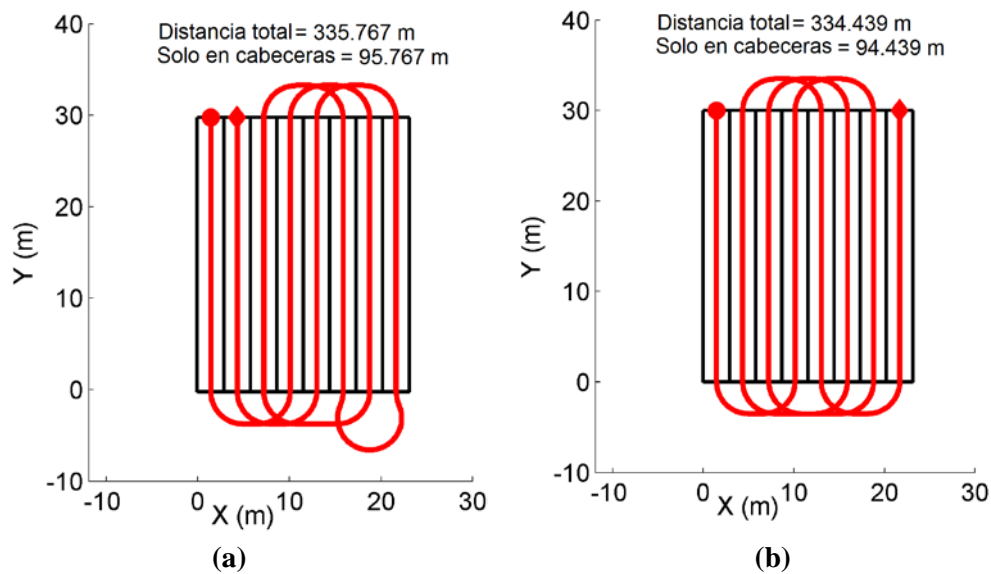


Figura 3.42. Rutas obtenidas para el primer campo, según la solución de (a) Bochtis y (b) el Planificador propuesto.

Los tres campos restantes son parte de una planificación conjunta en la que existe una transición mediante caminos entre campos. En el artículo (Bochtis & Vougioukas, 2008) no hay información sobre los caminos de conexión entre los campos por lo que se optó por comparar el funcionamiento del Planificador desarrollado individualmente con cada uno de los campos ya que los autores sí referían en el artículo las características de estos. Se configuró el Planificador para optimizar cada campo por separado considerando los mismos puntos de entrada y salida al campo reseñados en el artículo, es decir, determinados por las pistas de entrada y salida usadas en las soluciones proporcionadas por los autores. A saber, en el primer campo (campo2) de 20 pistas de 80 m de longitud se indicó al Planificador que empezara en la pista 20 y acabara en la pista 18. En el segundo campo (campo3) de 12 pistas de 40 m el Planificador empezó en la pista 1 y acabó en la pista 12. Por último, en el tercer campo (campo4) de 12 pistas de 70 m, se empezó por la pista 2 y se acabó en la pista 1. El ancho de las pistas era el mismo para los tres campos, 3 m, y el radio de giro del vehículo se fijó en 2,5 m.

Las Figuras Figura 3.44 y Figura 3.44 visualizan de izquierda a derecha las rutas de las soluciones obtenidas por (Bochtis & Vougioukas, 2008) y las rutas de las soluciones proporcionadas por el Planificador desarrollado para los campos campo2, campo3 y campo4.

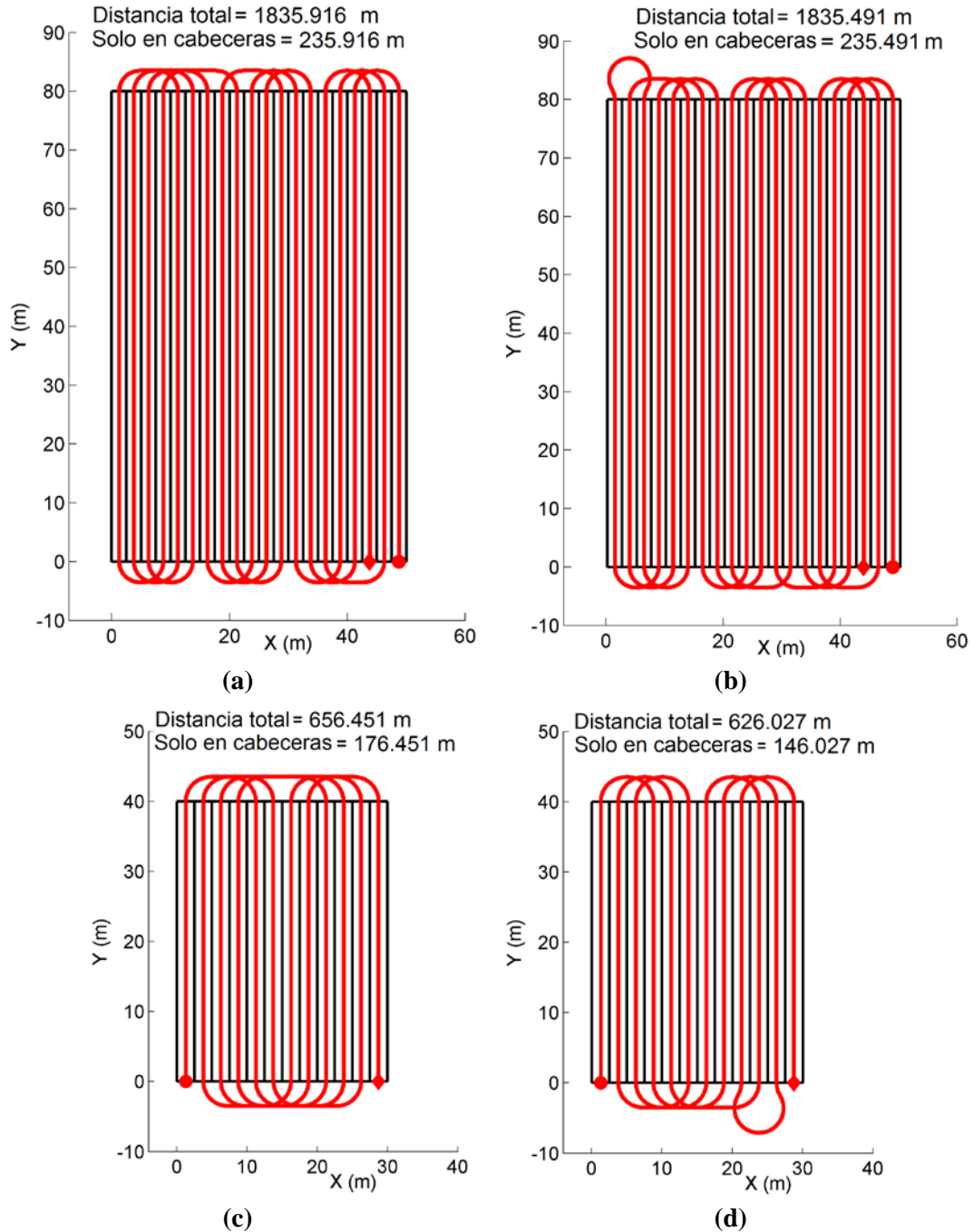


Figura 3.43. Soluciones obtenidas por (Bochtis & Vougioukas, 2008) para (a) el campo2 y (c) campo3. Rutas obtenidas por el Planificador desarrollado para (b) el campo2 y (d) campo3.

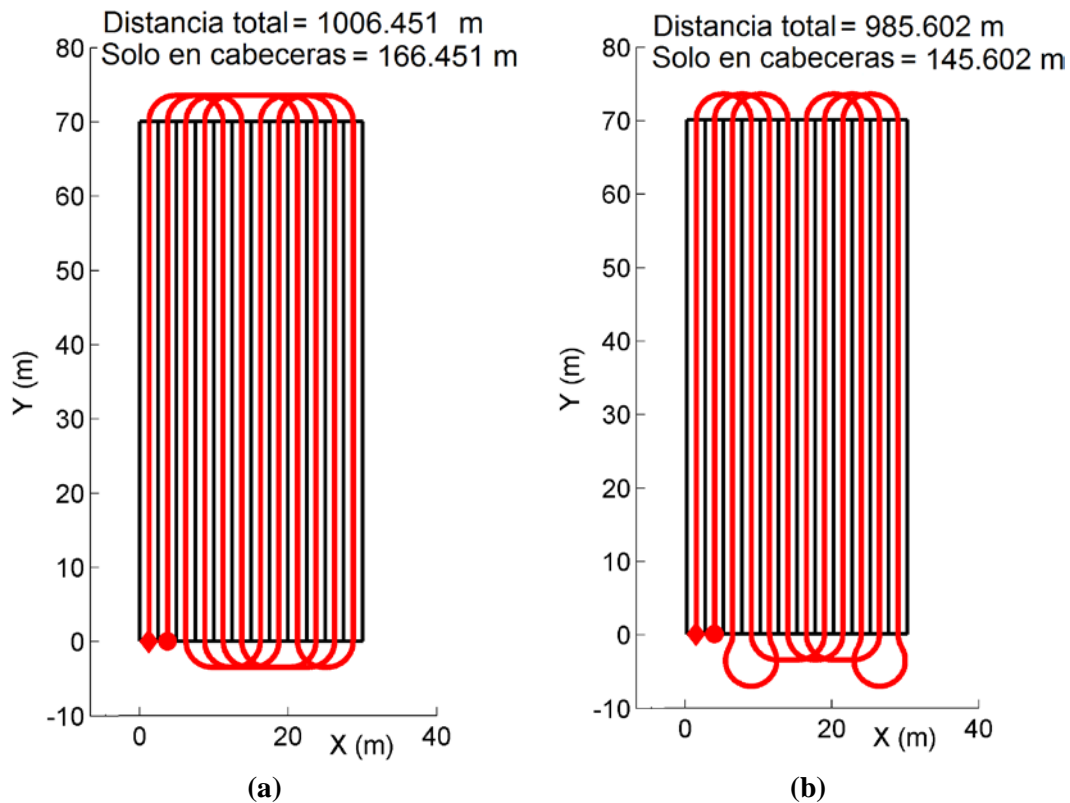


Figura 3.44. Soluciones obtenidas para el campo4 por (a) (Bochtis & Vougioukas, 2008) y (b) el Planificador desarrollado

Las tablas Tabla 3.29 y Tabla 3.30 contienen las soluciones y las distancias obtenidas por el Planificador desarrollado en este trabajo de investigación y por la aproximación de (Bochtis & Vougioukas, 2008). Todas las soluciones encontradas por el Planificador superan a las de la aproximación de (Bochtis & Vougioukas, 2008). En definitiva si se utilizasen los recorridos propuestos por el Planificador desarrollado añadiendo los mismos caminos de conexión entre campos la solución global sería mejor que la encontrada en (Bochtis & Vougioukas, 2008).

Tabla 3.29. Soluciones (vectores permutación) obtenidas por ambos métodos para cada campo.

Campo	Solución obtenida por Bochtis	Solución obtenida por el Planificador propuesto
campo1	(1,4,7,3,6,8,5,2)	(1,4,7,3,6,2,5,8)
campo2	(20,17,14,11,8,12,9,3,6,2,,5,1,4,7,10,13,16,19,15,18)	(20,17,14,11,8,5,2,6,3,1,4,,7,10,13,9,12,16,19,15,18)
campo3	(1,5,11,7,3,9,2,6,10,4,8,12)	(1,4,10,7,3,6,2,5,8,11,9,12)
campo4	(2,6,10,4,8,12,9,3,7,11,5,1)	(2,5,3,6,9,12,10,7,11,8,4,1)

Tabla 3.30. Distancias obtenidas por las soluciones obtenidas por ambos métodos (véase Tabla 3.29) para cada campo.

Campo	Resultados obtenidos por Bochtis		Resultados obtenidos por el Planificador propuesto	
	Distancia total (m)	Distancia en cabecera (m)	Distancia total (m)	Distancia en cabecera (m)
campo1	335,767	95,767	334,439	94,439
campo2	1.835,916	235,916	1.835,491	235,491
campo3	656,451	176,451	626,027	146,027
campo4	1.006,451	166,451	985,602	145,602

Por último el Planificador propuesto se comparó con la aproximación propuesta por (Hameed, Bochtis, & Sorensen, 2011), también basada en un método meta-heurístico. En este trabajo, los autores proponen un Algoritmo Genético para 1) determinar la mejor dirección de cultivo con la que establecer el campo, y 2) obtener la secuencia de pistas más corta para un único vehículo. Puesto que en la presente tesis la dirección de cultivo se considera como un parámetro de entrada puesto que en la etapa de tratamiento el cultivo ya está establecido, la comparación se llevó a cabo solo con respecto a las soluciones que aportan los autores en la optimización de la secuencia de pistas con objeto de encontrar las rutas de mínimo recorrido.

Uno de los experimentos considerados en el estudio de (Hameed, Bochtis, & Sorensen, 2011) contenía información suficiente para ser replicado. El vehículo tenía un radio de giro de 6 m. El campo estaba formado por 37 pistas de 9 m de ancho. El campo tenía forma irregular por lo que los puntos de entrada y salida de cada pista se extrajeron cuidadosamente a partir de Google Maps utilizando las coordenadas reales de la ubicación del campo, información que sí figuraban en el artículo. Con las coordenadas extraídas manualmente la suma total de las longitudes de las pistas era igual a 6.684 m, muy próxima a los 6.700,13 m referidos por los autores, además la forma de ambos campos encajaba con bastante precisión.

La solución proporcionada por los autores (véase Figura 3.45a) es (12,8,6,9,1,3,5,2,4,7,11,10,18,23,20,19,17,21,24,22,16,13,14,15,25,31,29,32,30,27, ..., ...,28,26,36,34,33,35,37), con una distancia total de 7.918,13 m y una distancia en cabecera de 1.218,00 m, mientras que la solución obtenida por el Planificador propuesto (véase Figura 3.45b), considerando una flota con un único vehículo y un campo libre de malas hierbas, fue (29,27,25,23,21,19,17,15,13,11,9,7,5,3,1,2,4,6,8,10,12,14,16,18, ..., ...,20,22,24,26,28,30,31,33,32,34,35,37,36), con una distancia total recorrida de 7.661,60 m y una distancia en cabecera de 961,47 m. Además, los autores se comparan en este artículo con el trabajo anterior de (Bochtis & Vougioukas, 2008) que obtiene para este problema una distancia recorrida en cabecera de 1.017,00 m; por lo que podemos concluir a la luz de los resultados, que el Planificador propuesto mejora tanto a la aproximación propuesta por (Bochtis & Vougioukas, 2008) como a la propuesta por (Hameed, Bochtis, & Sorensen, 2011).

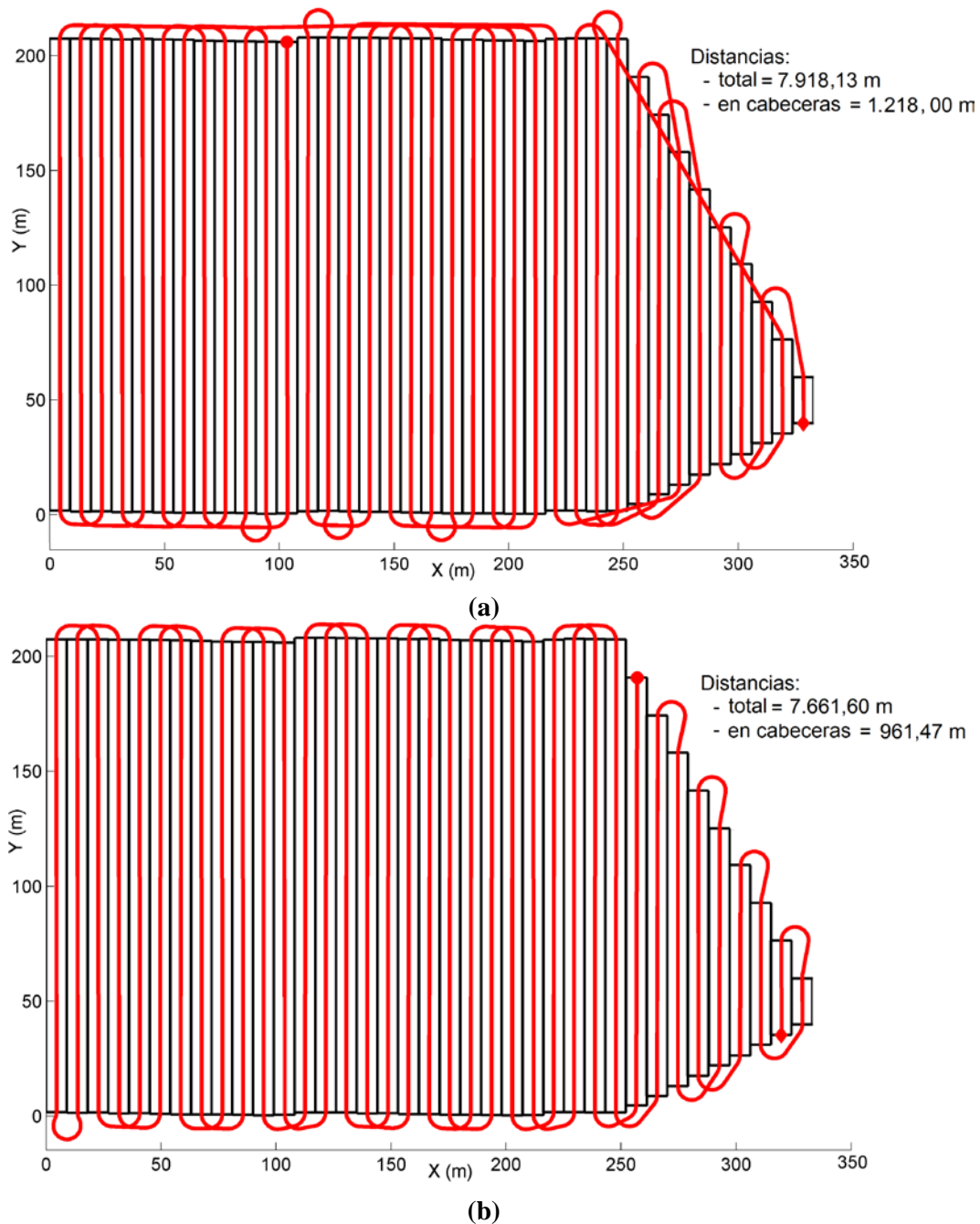


Figura 3.45. Comparación de las rutas obtenidas para el campo extraído de (Hameed, Bochtis, & Sorensen, 2011). (a) Ruta obtenida en el artículo de referencia y (b) ruta generada por el Planificador propuesto y desarrollado en este trabajo de investigación.

Finalmente, el Planificador desarrollado obtuvo la solución en aproximadamente 60 s, mientras que la aproximación propuesta por (Hameed, Bochtis, & Sorensen, 2011) necesitó 11,45 minutos para generar la solución.

4 SUPERVISIÓN DE MISIONES

Una vez tratado el tema de cómo planificar una tarea de forma óptima, el siguiente paso consiste en llevarla a cabo autónomamente y comprobar que se realiza según lo planeado. La parte concerniente a la robotización y automatización mecánica de la maquinaria a bajo nivel se escapa a los objetivos de esta tesis y se explica en el trabajo (Emmi, Gonzalez-de-Soto, Pajares, & Gonzalez-de-Santos, 2014), pero no así la supervisión o comprobación a alto nivel de que la tarea se está ejecutando correctamente.

El comportamiento de un sistema autónomo puede desviarse de lo esperado por la propia complejidad del sistema y, sobre todo, por la incertidumbre de los elementos externos que le afectan, muy impredecibles y difícilmente controlables, especialmente cuando se trabaja en exteriores. Es de vital importancia advertir cuanto antes las situaciones con malos comportamientos para poder reconducirlas o incluso detenerlas, sobre todo cuando los sistemas controlan maquinaria móvil grande y pesada como la agrícola y, por tanto, potencialmente peligrosa. Este capítulo trata sobre esta cuestión. En él se propone un sistema de supervisión distribuido con varios niveles, capaz de monitorizar la ejecución de la tarea, analizar lo sucedido, informar al operario responsable de la misión de los fallos o situaciones inesperadas que vayan apareciendo e, incluso en algunas ocasiones, subsanar el problema. En resumen, un sistema capaz de desempeñar los tres objetivos fundamentales de la supervisión: detección, diagnóstico y reparación.

La estructura del capítulo es la siguiente. La primera sección explica la arquitectura software que se propone para estructurar el sistema de supervisión. La siguiente sección detalla la instrumentación disponible para darle soporte físico. A continuación se explica el sistema que se implementó a partir de la arquitectura propuesta y de la instrumentación disponible. Y, por último en la sección final, se describen las pruebas que se llevaron a cabo para evaluar el sistema desarrollado y se discuten los resultados obtenidos.

4.1. Arquitectura de supervisión

Diseñar un sistema que sea capaz de monitorizar una flota de vehículos autónomos agrícolas mientras llevan a cabo una misión de forma cooperativa no es una tarea sencilla. Hay que supervisar una gran cantidad de aspectos (trayectorias, velocidades, aperos, etc.), lo que obliga a recibir y procesar mucha información en tiempo real. Solo así se puede

garantizar que el sistema sea capaz de reaccionar a tiempo a los posibles contratiempos que puedan surgir durante la misión, manteniendo la seguridad de la flota, de las personas que puedan estar próximas y del propio cultivo.

La gran cantidad de datos a procesar y el breve tiempo del que se dispone hace que ya desde el diseño deba pensarse en un sistema especialmente preparado para resolver cuanto antes las incidencias. Por ello, aprovechando que la misión debe ejecutarse de manera distribuida, ya que participan varios vehículos con sistemas de control independientes, parece conveniente usar un sistema de supervisión distribuido de tal modo que la supervisión exclusiva de un vehículo se ejecute en los equipos de a bordo, mientras que la supervisión que debe vigilar las situaciones que involucran a varios vehículos a la vez (por ejemplo, situaciones con una potencial colisión), se lleven a cabo en un equipo externo con acceso a los datos de todos los vehículos de la flota.

Distribuir el trabajo entre varias unidades de procesamiento, en teoría, debería acelerar en tiempo el funcionamiento del sistema, especialmente si se compara con soluciones monolíticas en las que un único sistema se encarga de procesarlo todo. Sin embargo, hay que ser cuidadoso ya que para coordinar las diferentes partes del sistema son necesarios mecanismos de comunicación que mal empleados pueden degradar mucho el rendimiento de todo el sistema, por ejemplo, por la aparición de interbloqueos (Raynal, 2013).

Para coordinar la supervisión distribuida se propone una arquitectura basada en pequeños módulos supervisores que actúan como agentes al servicio de un objetivo global (García-Pérez, García-Alegre, Ribeiro, & Guinea, 2008) y en alarmas (Wilkins, Lee, & Berry, 2003). Las alarmas son notificaciones que surgen cuando se detecta un fallo (velocidad inapropiada, vehículo fuera de trayectoria, boquilla con apertura incorrecta, etc.) o la consecución de un hito importante (cumplimiento de una trayectoria, arranque apropiado de un dispositivo, etc.). Los supervisores son pequeños bloques o módulos de procesamiento que generan alarmas al detectar el fallo o hito importante analizando la información periódicamente recibida desde los elementos supervisados, que pueden ser elementos físicos tales como motores, tanques, boquillas, sensores, etc., o elementos conceptuales como trayectorias, colisiones o incluso otros supervisores.

Mediante estos dos conceptos, alarmas y supervisores, es fácil desacoplar la supervisión en módulos independientes y fáciles de implementar. Además, puesto que los supervisores pueden emplearse para monitorizar otros supervisores, es posible construir incrementalmente un gran sistema global de supervisión en varias capas, desarrollando primero niveles bajos dedicados a la monitorización de los comportamientos más sencillos y, después, niveles más altos dedicados al control de los comportamientos más complejos que sean fruto de la combinación de los primeros. Por ejemplo, una misión con varios vehículos puede supervisarse a partir de la supervisión de cada uno de los vehículos por separado y, a su vez, un vehículo puede supervisarse a partir del comportamiento de cada uno de sus subsistemas internos; de ahí que la arquitectura propuesta también separe la supervisión en diferentes niveles o capas de menor a mayor alcance y complejidad.

Gracias a estas dos ideas, desacoplamiento (o distribución) y separación en niveles, la arquitectura propuesta parece un marco de trabajo útil para desarrollar fácilmente el sistema de supervisión requerido.

A continuación se detallan los niveles de supervisión propuestos para la arquitectura planteada.

4.1.1. Niveles de supervisión

La Figura 4.1 muestra los tres niveles propuestos para una flota de vehículos terrestres. La complejidad de la supervisión de cada nivel es creciente, es decir, cuanto menor es el nivel, más sencilla y más focalizada es la monitorización de los supervisores.

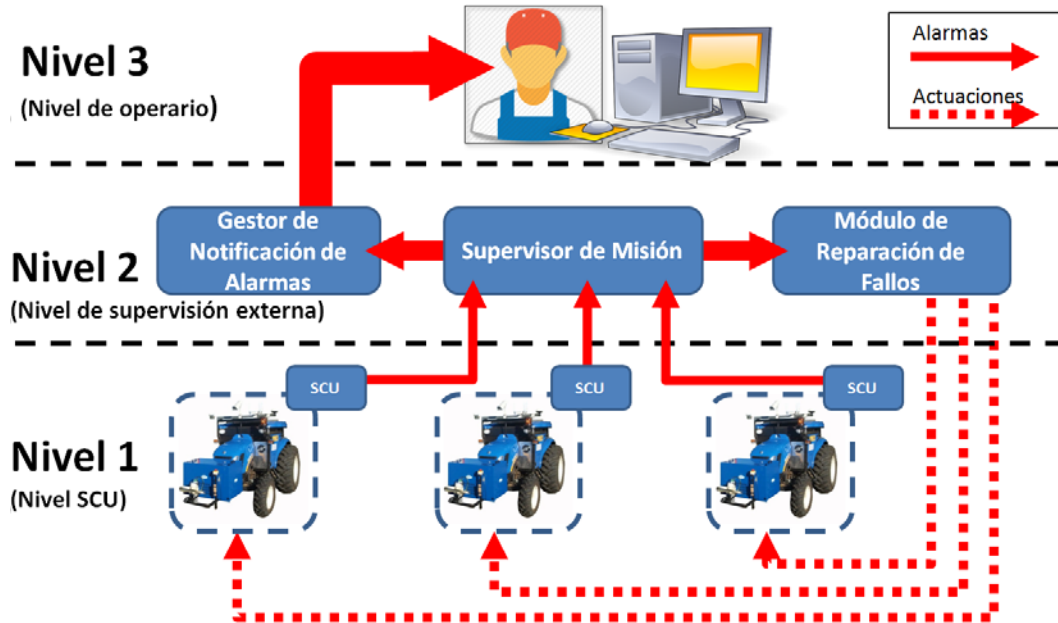


Figura 4.1. Supervisión distribuida y multi-nivel

El primer nivel se ejecuta dentro de los vehículos y básicamente está compuesto por los supervisores integrados en el Sistema de Control de la Unidad (SCU), que gobierna el vehículo y los dispositivos de percepción y actuación con los que está equipado. El desarrollo de un SCU es parte del proceso de robotización y automatización a bajo nivel de la maquinaria y, por tanto, su diseño y desarrollo están fuera de los objetivos de esta tesis, no obstante, es un sistema que debe conocerse porque sobre él se integran los supervisores de nivel más bajo de la arquitectura propuesta. Por lo general, los SCUs son capaces de informar de fallos básicos, como que un dispositivo no arranca, o situaciones relevantes, como que el nivel de combustible está bajo. A veces incluso pueden resolver malos comportamientos, por ejemplo, reiniciar un dispositivo que no está funcionando, sin embargo, aun en esos casos, siempre deben notificar los problemas detectados a los niveles de supervisión superiores mediante alarmas, para que las incidencias puedan analizarse desde una perspectiva global, ya que a veces los fallos pueden ser más significativos de lo que aparentan si, por ejemplo, se comprueba que se producen con frecuencia o vienen acompañados de otras anomalías.

Como los módulos de supervisión integrados en el SCU pueden ser capaces de detectar, diagnosticar y reparar, ya en este primer nivel se realizan las tres funciones principales de la supervisión.

Además de generar alarmas, este nivel también es responsable de propagar periódicamente la información proveniente de los sensores (posición GPS, velocidad del

vehículo, presión de una válvula, etc.) a las capas superiores para que en ellas se pueda utilizar y seguir con precisión el estado de las unidades. Para diferenciar esta información de la proporcionada por las alarmas, se la llamará información *de monitorización* de ahora en adelante.

En el segundo nivel o nivel de supervisión externa se detectan y diagnostican alarmas de más alto nivel, más complejas, y ya desde una perspectiva más global porque involucra a más de un elemento de la flota. Para ello, a diferencia de lo que sucede en el primer nivel, los módulos de este nivel se ejecutan en un equipo externo a las unidades, con acceso en tiempo real a la información de toda la flota, y con las tres funciones principales de la supervisión separadas en módulos más independientes. En concreto, el módulo Supervisor de Misión se dedica a la detección y el diagnóstico, y el Módulo de Reparación de Fallos a la reparación. Además existe un tercer módulo, el Gestor de Notificación de Alarmas, dedicado a filtrar las alarmas que deben enviarse al tercer nivel de supervisión, ya que, como se verá más adelante, es conveniente no enviar al operario toda la información posible para evitar saturarle y perjudicar su nivel de atención.

Por su importancia, el Supervisor de Misión, el Módulo de Reparación de Fallos y el Gestor de Notificación de Alarmas se explican por separado y con más detalle en las próximas secciones.

El tercer nivel o nivel de operario integra una interfaz gráfica de usuario (GUI, por sus siglas en inglés) que es la encargada de trasladar la información al operario a cargo de la misión. La GUI visualiza las alarmas que se reciben para que el operario disponga de la información necesaria para seguir con detalle el desarrollo de la misión. Además, permite al operario ser el último elemento de decisión y tomar control de la flota para cambiar directamente la propia misión si éste considera que algo no está funcionando según lo previsto; de ahí que, incluso en este nivel, también estén presentes las tres funciones principales de la supervisión: detección, diagnóstico y reparación.

4.1.2. Supervisor de Misión

Como ya se ha comentado, el Supervisor de Misión procesa la información facilitada por las unidades durante la misión y lleva a cabo una vigilancia de más alto nivel que la que se realiza en los vehículos. Este módulo está constituido por un conjunto de supervisores independientes trabajando juntos, cada uno de ellos dedicados en exclusiva a monitorizar un determinado aspecto, por ejemplo, la velocidad, las trayectorias o el estado del aparo de cada vehículo. De esta manera desacopla la supervisión en diferentes bloques de tal modo que es muy fácil desarrollar, sustituir, actualizar o añadir nuevas comprobaciones a la supervisión. Asimismo, el desacoplamiento permite ejecutar solo algunos módulos, lo que puede resultar interesante, por ejemplo, si alguno no está funcionando correctamente y está generando falsas alarmas o si alguno está sobrecargando el sistema con un procesamiento demasiado exhaustivo y no excesivamente importante, a saber, notificaciones continuas de errores pequeños de posición o velocidad. Más aún, como las salidas de los supervisores pueden conectarse como entradas de otros supervisores, al igual que con la arquitectura general propuesta, es posible combinarlos para implementar supervisiones de comportamientos mucho más complejos.

Para facilitar y hacer más comprensibles estas combinaciones, similarmente a cómo se hizo con la arquitectura general, también el Supervisor de Misión o nivel de

supervisión externa organiza los supervisores individuales en varios niveles. En concreto se proponen tres niveles: básico, unidad y de flota (véase Figura 4.2).

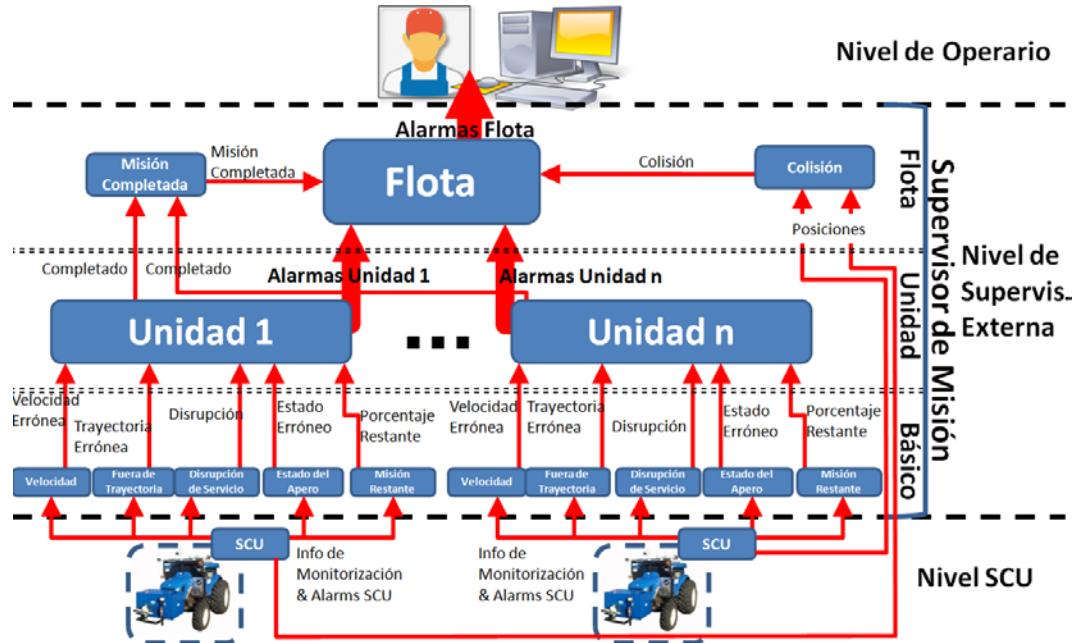


Figura 4.2. Arquitectura de supervisión donde se muestra el Supervisor de Misión (nivel 2) en detalle. Los rectángulos representan los supervisores del nivel 2 organizados jerárquicamente y las flechas las entradas y las salidas, donde los mensajes pueden ser tanto alarmas como información de monitorización.

Los supervisores del nivel básico reciben la información periódica asociada con alguno de los elementos de cada vehículo, tanto de los físicos (boquillas, sensores, etc.) como de los conceptuales (trayectorias, fin de la misión, etc.). Por consiguiente estos supervisores solo tienen la lógica para detectar los fallos relacionados con sus respectivos elementos, de ahí que sean básicos. Los del nivel de unidad detectan fallos que pueden surgir por anomalías en varios elementos de una misma unidad; por ejemplo, si un apero no puede usarse porque ninguna de sus boquillas funciona o cuando una unidad no tiene forma alguna de guiarse autónomamente porque fallan a la vez las cámaras y el GPS. Por último, los supervisores del nivel de flota vigilan que no se produzcan anomalías que involucren a más de un elemento de la flota, por ejemplo, colisiones entre varias unidades.

Además de la información proporcionada por los SCU del nivel inferior, el Supervisor de Misión también puede acceder a los datos de la misión especificados en el plan de actuación, esto es, a las trayectorias, velocidades y estados de los aperos previamente planificados. Esta información es estática (no cambia durante la ejecución de la misión) y, por tanto, puede ser cargada por cada supervisor al principio de cada ejecución.

La lógica interna de cada supervisor encapsula la funcionalidad de detección mientras que el diagnóstico va implícito en las propias alarmas, ya que cada una está asociada con un tipo concreto de fallo.

Por último, el Supervisor de Misión propaga tanto las alarmas generadas por él como las recibidas de los niveles inferiores al Módulo de Reparación de Fallos y al Gestor de Notificación de Alarmas.

4.1.3. Módulo de Reparación de Fallos

El Módulo de Reparación de Fallos es la entidad a cargo de neutralizar las averías o situaciones peligrosas informadas por las alarmas. Encapsula, por tanto, la funcionalidad de reparación. Recibe las alarmas emitidas por el Supervisor de Misión y accede a una base de datos donde está almacenado el protocolo de actuación o estrategia pre-establecida para resolver ese tipo de fallo. El protocolo consiste en una serie de órdenes para ejecutar en los SCUs. Por ejemplo, si una futura colisión entre varias unidades es detectada, la estrategia de neutralización puede consistir en ordenar la parada de alguna de las unidades para evitar la colisión.

Los protocolos de actuación están limitados por las órdenes que admiten los SCUs y que dependen del grado de autonomía con el que estén dotadas las unidades. Más adelante, en el apartado 4.2.3 se enumeran las órdenes que admiten los SCUs de la flota real utilizada para implementar el sistema propuesto en este trabajo de investigación.

4.1.4. Gestor de Notificación de Alarmas

El Gestor de Notificación de Alarmas es el conjunto de reglas que deciden qué alarmas deben propagarse hasta el operario (nivel 3 en la Figura 4.1) basándose en una política que tiene en cuenta la repetición y la gravedad de las alarmas. Como se explica en (Wilkins, Lee, & Berry, 2003), saturar de información al operario puede ser contraproducente porque puede llegar a no dar a las alarmas la importancia que se merecen o incluso a ignorarlas. Es necesario, por tanto, filtrar la información que se envía y no comunicar absolutamente todo lo que sucede. Por ejemplo, si se dispone de una herramienta termo-mecánica para eliminar malas hierbas mediante pequeñas llamas controladas, puede suceder que la llama se extinga de forma prematura por viento o cualquier otra condición atmosférica adversa, generando una alarma que a su vez desencadena una reparación por parte del Módulo de Reparación de Fallos que da lugar a un nuevo proceso de ignición. Este proceso de apagado y encendido puede repetirse con mucha frecuencia si las condiciones meteorológicas así lo propician y entonces no sería necesario estar informando constantemente de lo mismo al operario, a no ser que las interrupciones fueran tan frecuentes que impidieran realizar el trabajo y que requiriese la presencia de un operario en campo para evaluar la operatividad del equipo y repararlo en caso necesario.

Por otro lado, también puede suceder que una alarma esté permanentemente activada y, aunque entonces sí sea necesario notificar al operario para que tome las medidas oportunas, no convenga hacerlo con la misma frecuencia con la que se recibe, sino con cierto espacio de tiempo para no abrumarle y así no distraerle de otros sucesos que puedan ser más relevantes.

Estos casos son ejemplos en los que filtrar alarmas parece una práctica interesante, no obstante, para prevenir cualquier pérdida de información, este módulo también debe guardar todas las alarmas generadas durante la misión (por ejemplo, en un fichero o base de datos). Estas trazas permitirán al operario la consulta a posteriori o, incluso, que un

sistema de obtención de patrones (*data mining*) analice posibles patrones de comportamiento en la ejecución de las misiones que ayuden a mejorar la eficacia de la flota.

4.2. Instrumentación

Antes de describir en detalle la implementación del sistema de supervisión bajo la arquitectura propuesta, es necesario conocer los dispositivos (sensores, vehículos, aperos, etc.) de la que se dispuso como soporte físico. Este equipo determinó en gran medida la información con la que se podía trabajar y, por tanto, el propio sistema de supervisión desarrollado. En concreto se contaba con los dispositivos y la flota de vehículos terrestres de RHEA (RHEA, 2010-2014). En la Figura 4.3 se muestra el esquema general del proyecto RHEA. Además de la flota, había una estación base con un ordenador principal para manejar remotamente los vehículos y ejecutar algunos servicios tales como la construcción de mapas de malas hierbas a partir de imágenes aéreas tomadas con drones o generación de planificaciones basadas en la aproximación explicada en el Capítulo 3. El ordenador de la estación base fue el elemento escogido para ejecutar la parte del sistema de supervisión externo a las SCUs, es decir, los niveles 2 y 3 de la Figura 4.1. Sus componentes eran: placa base ASUS Z87-K SK1150/PCX 3.0, procesador Intel core i7 4771 3,5 GHZ, dos módulos de memoria DDR3 1600 8GB PC3-12800 (en total 16 GB RAM) y un dispositivo de almacenamiento de estado sólido (SSD) de 240 GB. Con estas características, el ordenador tenía la potencia suficiente para ejecutar simultáneamente el Supervisor de Misión, el Módulo de Reparación de Fallos y el Gestor de Notificación de Alarmas así como software adicional como la interfaz gráfica de usuario (GUI).



Figura 4.3. Arquitectura del proyecto RHEA

Respecto a las unidades, la flota RHEA estaba formada por tres tractores autónomos que podían configurarse para usar varios aperos. Cada vehículo autónomo se había desarrollado a partir de un tractor de tamaño medio, New Holland Boomer 3050 (Boomer 3000 series, 2015), de 50 CV y 1.270 kg, al que se le había cortado la cabina (véase Figura 4.4) integrando en el espacio libre algunos equipos como el ordenador de a bordo, el receptor GPS, el controlador de bajo nivel, etc. Además de en la cabina, otra parte de los dispositivos se integraron en el exterior del vehículo, a saber, una cámara RGB, un láser, tres antenas, (dos del receptor GPS y una para comunicaciones), luces de emergencias, una pila de combustible y un panel solar situado en el techo.

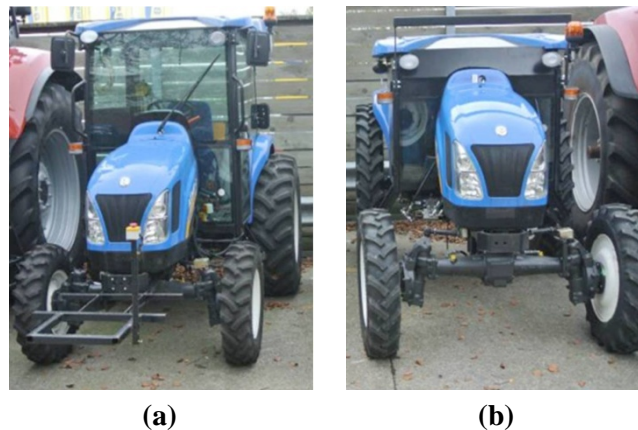


Figura 4.4. Cabina del tractor antes (a) y después de la adaptación (b).

En la Figura 4.5 se muestran las tres unidades tras su automatización, distinguiendo con claridad tres partes fundamentales: el propio tractor con el sistema de control o SCU en su interior (5), los dispositivos de percepción (1) y las tres herramientas de actuación o aperos (2, 3 y 4).



Figura 4.5. Flota de RHEA: (1) Sistemas de percepción, (2) barra de pulverización, (3) herramienta mecánico-térmica, (4) barra de pulverización de inyección directa y (5) sistemas de control.

En las siguientes secciones se habla de los diferentes dispositivos empleados dentro de cada una de estas partes, prestando especial atención a aquellos que se utilizaron para proporcionar información a la supervisión.

4.2.1. Percepción: Sensores

Para percibir los cambios en el entorno de trabajo, los vehículos estaban equipados con multitud de sensores (Emmi, Gonzalez-de-Soto, Pajares, & Gonzalez-de-Santos, 2014), entre los que destacan cuatro: una cámara, una unidad de medida inercial (IMU - Inertial Measurement Unit) solidaria a la cámara, un receptor GPS-RTK y un sistema LiDAR (Light Detection and Ranging). Todos ellos pueden verse en la Figura 4.7.

La cámara, modelo SVS4050CFLGEA de SVS-VISTEK, con sensor CCD Kodak KAI 04050M/C con filtro de color GR Bayer (véase Figura 4.6), suministraba imágenes de alta resolución, en concreto 2,336 x 1752 píxeles con un tamaño de pixel de 5.5 x 5.5 μm , a partir de las cuales se localizaban las malas hierbas y las líneas de cultivo.



Figura 4.6. Caja con cámara, IMU, termostato y ventilador

La IMU (véase Figura 4.6), modelo 3DM-GX3[®]-35 desarrollado por LORD MicroStrain[®] Sensing Systems, proporcionaba información precisa sobre el movimiento al que se veía sometida la cámara debido al desplazamiento del tractor en terreno irregular; en otras palabras suministraba los ángulos de guiñada (*yaw* en inglés), cabeceo (*pitch*) y alabeo (*roll*).

Como el equipo trabajaba en exteriores tanto la cámara como la IMU se situaron dentro de una caja refrigerada con un ventilador conectado a un termostato (véase Figura 4.6). Gracias a este sistema los dispositivos podían trabajar incluso con lluvia o con temperaturas de 50°C.

El receptor GPS de tipo RTK (del inglés *Real Time Kinematic*) de cada tractor era un modelo Trimble BX982 GNSS (véase Figura 4.7), multi-canal y multi-frecuencia capaz de proporcionar la localización con un error de centímetros. Se empleó un modelo de dos antenas que calculaba, además, la orientación de los vehículos con respecto al punto central del eje que unía las dos antenas. En concreto el receptor GPS-RTK era

capaz de devolver la orientación del vehículo con una precisión de décimas de grado y la posición con una precisión relativa de $0,008 + 1 \text{ ppm}$ (partes por millón) en el plano horizontal, o lo que es lo mismo, por debajo de 2 cm y a una frecuencia máxima de 20 Hz.

Por último, los vehículos disponían también de un LiDAR (véase Figura 4.7) como medida extra de seguridad para detectar la aparición de obstáculos no previstos como personas, animales, árboles caídos, etc. Se usó un modelo LMS 111 (SICK AG, Walldkirch, Germany) que se instaló en la parte frontal central del vehículo con una leve inclinación de 4° sobre la horizontal y ajustado para detectar obstáculos en el sentido de la trayectoria del vehículo a unos 70 cm de altura. El LiDAR era el último elemento de seguridad anti-colisión de las unidades, por lo que estaba directamente conectado a la alimentación y los motores de la unidad, de tal modo que, en caso de detectar algún obstáculo que no hubiese detectado el sistema de visión, detenía el vehículo, siendo solo posible ponerlo en marcha de nuevo manualmente.

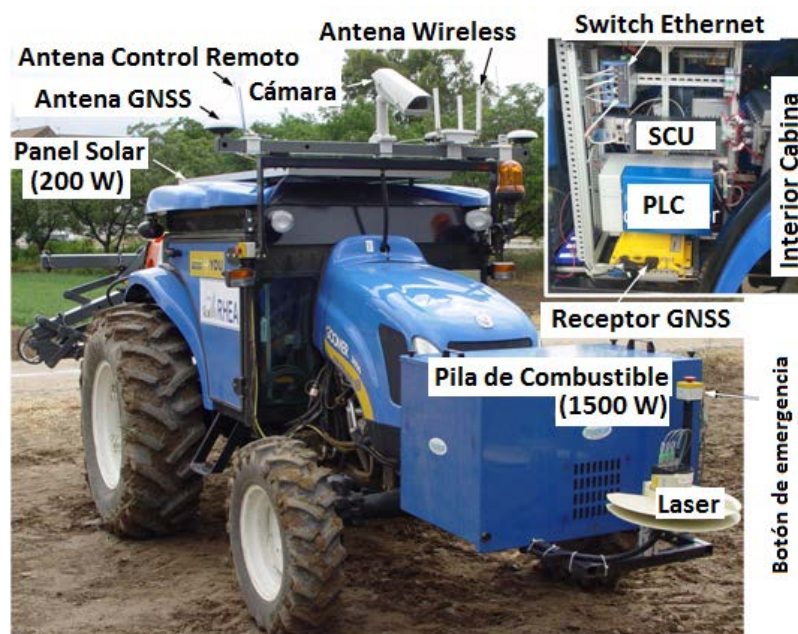


Figura 4.7. Dispositivos instalados en los tractores

4.2.2. Actuación: Aperos

En RHEA se usaron tres aperos que se podían accionar enviando ordenes desde el ordenador de a bordo del vehículo al Controlador Lógico Programable o PLC (en inglés *Programmable Logic Controller*). Esto aperos, que se describen en lo que sigue, eran: un equipo de tratamiento de mala hierba con herbicida, una herramienta mecánico-térmica y un equipo para rociar árboles.

El equipo de tratamiento de malas hierbas (véase Figura 4.8) fue desarrollada por Agrosap (Carballido, Perez-Ruiz, Gliever, & Agüera, 2012) e integraba dos tanques y una barra de 5,5 m y 12 boquillas distribuidas uniformemente, con un ancho trabajo cada una de 50 cm, por lo que el ancho de trabajo de la barra era de 6 m). Las boquillas podían

abrirse y cerrarse independientemente. El tanque principal tenía una capacidad de 200 L y el secundario de 15 L. Además, el tanque secundario contaba con un sistema de inyección directa para inyectar el producto químico directamente en la barra. Gracias a este sistema se mezclaba el herbicida con el agua justo unos instantes antes de la aplicación, evitando mezclar el herbicida y el agua con anterioridad al inicio del tratamiento, lo que requiere conocer de antemano la cantidad de herbicida a utilizar so pena de tener que hacer alguna parada para rellenar el tanque, es decir, encarecer la operación de tratamiento, o tener que desechar el producto sobrante con la consiguiente contaminación medioambiental. En definitiva, un equipo de inyección directa es el que mejor se adapta a un tratamiento localizado de mala hierba. Los tanques también contaban con sensores ópticos para saber en todo momento los volúmenes de agua y herbicida disponibles.

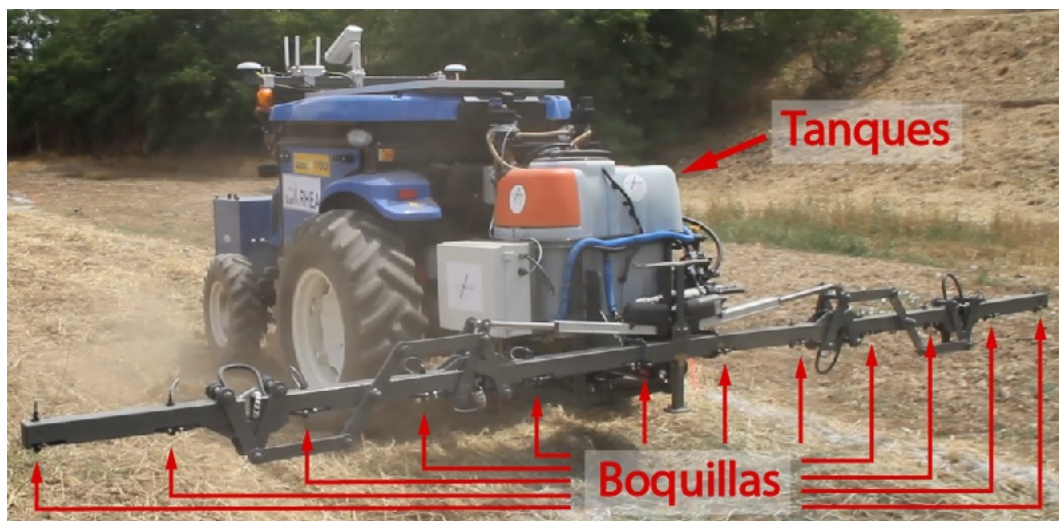


Figura 4.8. Equipo de pulverización de herbicida desarrollado por Agrosap para RHEA.

La herramienta mecánico-térmica (véase Figura 4.9) trabajaba sobre cuatro líneas de cultivo mediante cuatro pares de quemadores (un par por línea). Fue desarrollada por la universidad de Pisa (Perruzzi, Frascioni, Martelloni, Fontanelli, & Raffaelli, 2012)(Raffaelli, Martelloni, Frascioni, Fontanelli, & Perruzzi, 2013) para el control de malas hierbas en aquellos campos con cultivos resistentes a llamas pequeñas y poco intensas, como el maíz, la cebolla o el ajo. El tallo de estas plantas está recubierto por varias capas de hojas que actúan como un escudo cuando se les aplica una pequeña llamarada. Tras el golpe de calor las hojas externas se secan y caen, pero con el tiempo acaban siendo reemplazadas sin que la planta se resienta. Por el contrario, las malas hierbas, que no disponen de este mecanismo de protección, no pueden recuperarse y mueren.

Además de los quemadores, el apero tiene una parte mecánica, más tradicional, compuesta por una barra con cinco azadas pensada para eliminar las malas hierbas que crecen en el espacio entre líneas. Su ancho de trabajo de la parte mecánica era de cuatro calles (tres calles internas y media a cada lado). La barra se baja para que las azadas se hundan en el suelo y levanten la tierra con el movimiento del tractor. La acción mecánica es constante, es decir, se ejecuta siempre independientemente de que haya o no malas hierbas, mientras que la acción térmica es localizada y solo se activa cuando se detectan

malas hierbas; por lo que este aporo requiere un sistema de detección de malas hierbas en tiempo real. En el proyecto RHEA, este sistema se desarrolló analizando las imágenes RGB proporcionadas por la cámara del sistema de percepción (Montalvo, Guerrero, Romeo, Emmi, Guijarro, & Pajares, 2013)(Romeo, y otros, 2013)(Emmi, Gonzalez-de-Soto, Pajares, & Gonzalez-de-Santos, 2014) que no solo era capaz de detectar si había malas hierbas, sino que también estimaba la cantidad y la presión (baja o alta) con la que debían actuar los quemadores.

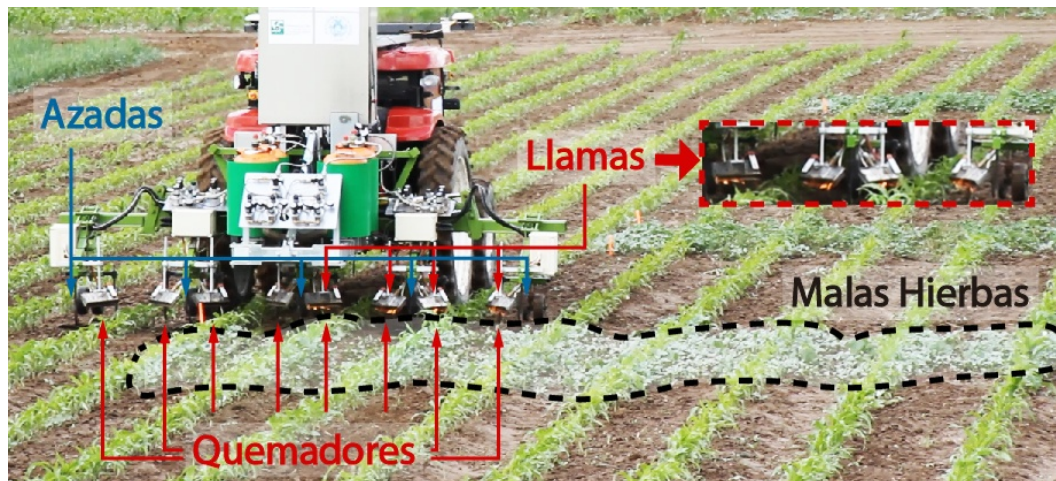


Figura 4.9. Herramienta mecánico-térmica para la eliminación precisa de malas hierbas.

Por último, el pulverizador de fitosanitario por aire comprimido (véase la Figura 4.10) estaba pensado para tratar árboles como olivos o frutales.

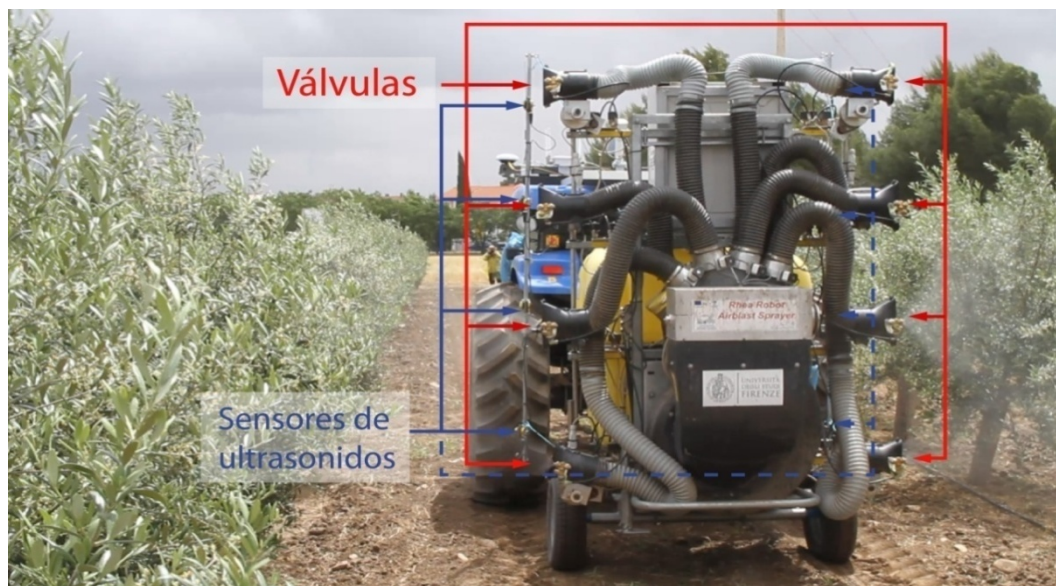


Figura 4.10. Pulverizador de fitosanitario por aire comprimido.

Esta herramienta fue desarrollado por la universidad de Florencia(Vieri, Lisci, Rimediotti, & Sarri, 2012),(Vieri, Lisci, Rimediotti, & Sarri, 2013). Estaba formado por 8 conductos acabados en 8 boquillas por las que sale el producto, 4 por cada lado y a diferentes alturas. Delante de las boquillas había dos barras de sensores de ultrasonidos, una a cada lado, para detectar si había follaje a la altura de la boquilla, y, en caso de no ser así, desactivar la emisión de aire comprimido por la boquilla para ahorrar fitosanitario. Además, las boquillas superiores e inferiores eran capaces de variar su dirección hasta 30° según dónde detectasen el follaje para ajustar la pulverización a la forma de la copa y conseguir, así, un efecto envolvente.

4.2.3. Control: Sistema de Control de Unidad

Además de los dispositivos de percepción y actuación, los tractores de RHEA estaban equipados con un ordenador central en el que se ejecutaba todo el software de control de la unidad (SCU). El ordenador era un CompactRIO 9082 de National Instruments, seleccionado por su sistema multi-core de alto rendimiento. En concreto las características de este modelo eran procesador Intel Core i7 dual-core de 1.33 GHz, con 32 GB de memoria no volátil y 2 GB DDR3 800 MHz de memoria RAM.

Es en este ordenador en el que se ejecuta el nivel 1 de supervisión de la Figura 4.1 y por tanto el que alberga el SCU. Este sistema es el que integra todos los dispositivos instalados a bordo y los coordina para que la unidad funcione autónomamente y se pueda manejar en remoto. Por otro lado, también genera las salidas de cada vehículo al exterior, es decir, la información de monitorización y las alarmas que proporcionaba este nivel (nivel 1 en la Figura 4.2) al nivel superior. La información de control y las alarmas que emite están enumeradas en las tablas Tabla 4.1 y Tabla 4.2, respectivamente. Estas salidas son importantes porque es la información que le llega al Supervisor de Misiones para realizar su labor.

Sobre el control en remoto de la unidad, la Tabla 4.3 lista con una breve descripción las principales órdenes que puede ejecutar el SCU para los dispositivos de a bordo incluido el vehículo. Estas órdenes son importantes porque son las acciones que puede llevar a cabo el Módulo de Reparación de Fallos para resolver problemas.

Tabla 4.1. Información de monitorización que proporciona periódicamente el SCU.

Sistema Origen	Tipo de Info	Descripción
Vehículo	Control	Indica el estado del controlador interno del vehículo: desactivado, activado, preparado, manual o ejecutando una misión, pausado o parado.
	Movimiento	Proporciona la posición GPS, la velocidad, la orientación del vehículo, las revoluciones del PTO y la altura TPH.
Cámara	Imágenes	Muestran lo que visualiza la cámara instalada en la parte frontal
Cualquier Apero	Apero	Indica el estado del apero (OFF/ON)
Herramienta Mecánico- Térmica	Quemadores	Informa de la presión de cada quemador (OFF/Baja/Alta)
Barra de Pulverización	Boquillas	Informa de la presión de cada boquilla (OFF/ON)

Tabla 4.2. Alarmas que puede generar el SCU para el nivel 2 (Supervisor de Misiones).

Sistema Origen	Alarma	Descripción
Equipo de a bordo	Temperatura de la placa base	El sistema debe parar para refrigerarse y evitar daños en el hardware
	Temperatura de CPU	El sistema debe parar para refrigerarse y evitar daños en el hardware
	Nivel memoria RAM	El sistema está consumiendo mucha memoria y puede ralentizarse
	Fallo de sincronización en comunicaciones	El tiempo del sistema no se ha sincronizado correctamente con el del GPS
Vehículo	Misión abortada por vehículo	El vehículo no se mueve por algún error interno
	Misión abortada por giro	El vehículo no puede realizar el giro indicado
	Misión abortada por petición externa	El vehículo se ha parado por petición expresa desde algún nivel externo
	Misión abortada por apero	El vehículo se ha parado por error en el apero
	Misión terminada	El vehículo se ha parado porque ha completado satisfactoriamente la misión encomendada.
Cámara	Fallo en cámara	Fallo en el funcionamiento de la cámara
PTO	Error en ajuste de PTO	El PTO no pudo ajustarse a las revoluciones indicadas
TPH	Error en ajuste de TPH	El TPH no pudo ajustarse a la altura indicada
Cualquier Apero	Error en Start/Stop	El apero no pudo activarse/desactivarse correctamente
Barra de Pulverización	Nivel en tanque principal	Nivel crítico en tanque principal
	Nivel en tanque secundario	Nivel crítico en tanque principal (tanque del sistema de inyección directa)
	Error en ajuste de flujo principal	El flujo del tanque principal no pudo ajustarse
	Error en ajuste de flujo secundario	El flujo del tanque secundario no pudo ajustarse
	Línea bloqueada	El conducto de herbicida está bloqueado
	Fallo en apertura/cierre de barra	La barra no pudo desplegarse/plegarse correctamente
	Fallo en apertura/cierre de boquilla	La boquilla no pudo abrirse/cerrarse correctamente
Herramienta Mecánico-térmica	GLP* agotado	La botella de GLP está vacía
	Quemador apagado	Se ha apagado un quemador que debería estar encendido
	Error en estado de quemadores	Los quemadores no pudieron ajustarse (OFF/bajo/alto) correctamente
Pulverizador de Aire Comprimido	Flujo insuficiente	No hay flujo suficiente en el conducto principal
	Nivel tanque	Nivel crítico en tanque de fitosanitario
	Error en sensor de válvula	El sensor de ultrasonidos de una de las válvulas no está funcionando correctamente

*GLP: Gas licuado del petróleo

Tabla 4.3. Ordenes más importantes que genera el SCU para los dispositivos de a bordo, incluido el vehículo

Dispositivo sobre el que actúa	Orden	Descripción
Vehículo	Start	Inicializa la unidad (fija el origen de coordenadas, limpia los registros, etc.
	Stop	Para la unidad sin posibilidad de reanudar la acción que se estaba llevando a cabo, porque limpia completamente los registros.
	Pausa	Para la unidad con posibilidad de reanudar la acción que se estaba llevando a cabo
	Reanudación	Reanuda la unidad para que continúe por la acción que estaba llevando a cabo antes de ser pausada.
	Mover	Mueve la unidad 5 m con una determinada velocidad y dirección
	Misión	Ejecuta un plan de actuación
	Modo	Cambia entre modo manual, para el control remoto de la unidades, y modo automático, comportamiento autónomo de las unidades para ejecutar misiones.
	Reset	Reinicia totalmente el SCU
PTO*	Set PTO	Ajusta las revoluciones del PTO
TPH**	SetTPH	Ajusta la altura del TPH
Cualquier Apero	Start/Stop	Activa/desactiva el apero
Herramienta Mecánico-Térmica	Set quemadores	Ajusta el nivel de presión de los quemadores
Barra de Pulverización	Set boquillas	Ajusta si las boquillas de la barra de pulverización están abiertas o cerradas
	Set DIS	Ajusta el sistema de inyección directa
	Set flujo	Ajusta el flujo principal
	Set dosis	Ajusta la cantidad de la dosis de herbicida que aplica cada boquilla

*PTO (power take-off): sistema para transferir energía del motor al apero

**TPH (three point hitch): pieza que engancha el apero al tractor

4.3. Sistema se supervisión implementado

En esta sección se describe cómo se implementaron los módulos principales del sistema de supervisión, a saber, el Supervisor de Misión, el Módulo de Reparación de Fallos y el Gestor de Notificación de Alarmas, así como la interfaz gráfica de usuario (GUI) basándose en los vehículos y dispositivos de disponibles en el proyecto RHEA.

4.3.1. Implementación del Supervisor de Misión

Como se ha comentado ya, el Supervisor de Misión está compuesto por múltiples módulos supervisores que se reparten la supervisión de alto nivel en bloques separados

que colaboran entre sí. Estos módulos se implementaron como pequeños bloques de código en forma de sistemas de reglas IF-THEN que generan alarmas cuando la información recogida cumple ciertas condiciones establecidas previamente.

El nivel básico contiene los supervisores que directamente trabajan con la información proporcionada por cada SCU. En concreto, los supervisores que se implementaron monitorizan la velocidad, la trayectoria, el estado de los aperos y cuanta misión o porcentaje queda por completar. Además se monitoriza que no haya interrupciones de servicio, o lo que es lo mismo si se está recibiendo periódicamente la información de monitorización desde el nivel SCU (véase Tabla 4.4).

El *supervisor de velocidad* recibe la velocidad, la posición y la orientación del vehículo con una frecuencia de 20 Hz, que coincide con la frecuencia de trabajo de los receptores GPS-RTK a bordo de los vehículos, y comprueba que la velocidad corresponde con la esperada en ese punto de la trayectoria, con cierto margen de tolerancia, y con esa orientación según lo especificado en el plan de actuación. En caso de que no sea así, genera una alarma. De forma muy parecida el *supervisor de fuera de trayectoria* comprueba cada décima de segundo que el vehículo se está moviendo por la trayectoria planeada; para ello verifica que cada posición que recibe encaja con la localización que se espera que tenga en ese momento el vehículo. En caso de que no se esté en el punto indicado, genera una alarma. Dado que el GPS-RTK que se empleó proporciona las posiciones con una precisión cercana a los 2 cm y a 20 Hz, y teniendo en cuenta que la velocidad máxima de trabajo que se consideró para los vehículos era de 10 km/h (aproximadamente 2,78 m/s) está garantizada la detección de un mal funcionamiento a tiempo de detener el vehículo sin causar daños, ya que a la máxima velocidad la comprobación de la posición se hace cada 13 cm.

El *supervisor de estado de apero* se encarga de comprobar que el estado de los actuadores (boquillas, quemadores, etc.) es el correcto para ese punto de la misión y genera una alarma en caso de no ser así. Por su parte, el *supervisor de misión restante* informa de qué porcentaje de la misión queda por completar en ese momento, por lo que más que generar una alerta por fallo, proporciona información extra de control para los niveles superiores. En concreto, cada vez que se recibe una nueva posición del vehículo, este supervisor calcula cuanta trayectoria queda por delante y emite una alarma con un porcentaje.

Por último, el *supervisor de interrupción de servicio* comprueba que el SCU proporciona periódicamente la información necesaria. Como se están supervisando tractores de tamaño medio en movimiento, hay cierta información crítica que es obligatorio recibir siempre con una mínima frecuencia para garantizar que se tiene todo bajo control. Para ello, este supervisor va registrando los tiempos de recepción de los mensajes que van entrando con información crítica, y emite una alarma en caso de que se sobrepasase una determinada cantidad de tiempo sin recibir alguno de ellos.

En el nivel de unidad se concentran todas las alarmas generadas por los supervisores del nivel básico en un único supervisor por vehículo llamado *supervisor de unidad*. De este modo se combina toda la supervisión proporcionada por los supervisores básicos, creando un supervisor robusto y completo de cada unidad, que controla un amplio catálogo de situaciones no deseadas. Este nivel tiene tantos supervisores como unidades haya en la flota.

Finalmente, en el nivel de flota se incluyeron tres supervisores: el *supervisor de colisiones*, el de *misión completada* y el de *flota*, que generan el catálogo de alarmas detalladas en la Tabla 4.5.

Tabla 4.4. Alarmas generadas por los supervisores de nivel básico.

Supervisor	Alarma	Descripción
Velocidad	Velocidad incorrecta	La unidad no se está moviendo a la velocidad esperada
Fuera de Trayectoria	Posición incorrecta	La unidad no está en la posición esperada
Disrupción de Servicio	Disrupción	No se está recibiendo la información de un sensor o algún dispositivo de la unidad con una frecuencia aceptable.
Estado de Apero	Estado incorrecto	El apero no está en el estado esperado (por ejemplo, una boquilla no está abierta)
Misión Restante	Porcentaje restante	Porcentaje de misión que queda por completar hasta ese momento.

Tabla 4.5. Alarmas del *supervisor de flota*.

Supervisor	Alarma	Descripción
Colisión	Muy cerca	Dos unidades están muy cerca
	Colisión	Dos unidades van a colisionar próximamente
	Camino libre	La unidad pausada ya tiene el camino libre para continuar con la misión
Misión Completada	Misión Completada	Todas las unidades han terminado la misión
Flota	Todas	Todas las alarmas generadas en este nivel y en cualquiera de los inferiores

El *supervisor de colisiones* analiza las próximas posiciones de las unidades según sus trayectorias esperadas y la posición real para detectar potenciales colisiones que se podrían producir si no se toman medidas. Es importante aclarar en este punto que en el caso de las misiones de tratamientos de malas hierbas en cultivos de calle ancha (véase Tabla 1.2) los planes sólo indican los puntos de entrada y salida del cultivo para garantizar la cobertura total del mismo y es el sistema de navegación basado en la detección de línea de cultivo (Emmi, 2014) el que guía al vehículo, asimismo los vehículos transitan libremente por un espacio común que son las cabeceras, éstas son las dos situaciones principales en las que es posible que se produzca un riesgo de colisión entre vehículos sin que estos estén funcionando mal. En definitiva, este supervisor hace un primer análisis en dos casos. En el primero comprueba dos a dos si las unidades están muy cerca, en concreto por debajo de una determinada distancia límite, y, en caso de ser así, lo notifica mediante una alarma. En caso contrario se pasa al segundo supuesto, en él se calculan las posiciones futuras de la flota moviéndolas incrementalmente según las trayectorias esperadas (Figura 4.11). Para ello es necesario avanzar las unidades usando pequeños intervalos de tiempo que garanticen que no se pueden colar colisiones entre posición y posición, para lo cual se debe tener en cuenta para todas las unidades la superficie que abarcan y la velocidad a la que se están moviendo, ya que cuanto mayor sea la velocidad más pequeño tendrá que ser el intervalo.

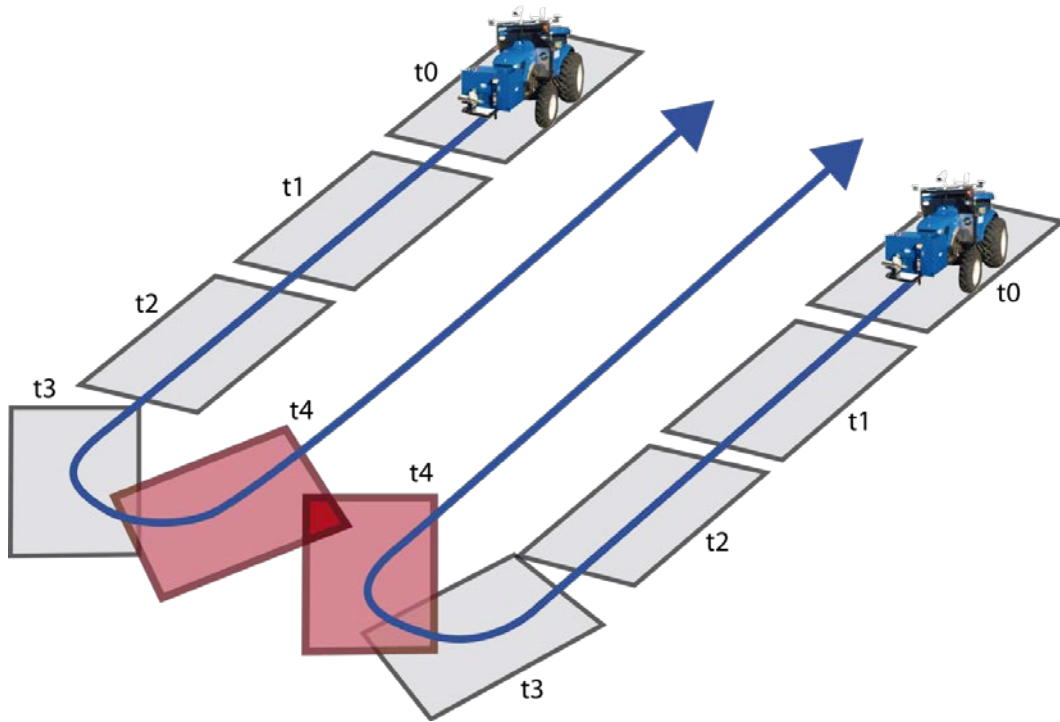


Figura 4.11. Avance incremental de las posiciones del *supervisor de colisiones*.

La superficie que ocupa la unidad se calcula en función de sus dimensiones reales incluyendo los aperos desplegados dentro del campo y recogidos en cabecera y añadiendo un margen de seguridad que tiene en cuenta distintos aspectos como el error del GPS-RTK (2 cm), que la unidad no frena instantáneamente o el propio tiempo de respuesta del supervisor condicionado por la sobrecarga del ordenador.

El margen lateral de seguridad debe ajustarse de tal modo que la anchura de la superficie nunca sea inferior a la anchura de la unidad, es decir, debe ser el máximo entre la anchura del vehículo y la del apero. Tampoco puede ser mayor que la anchura de pista, para que no detecte a unidades dentro de los límites de las pistas adyacentes. El margen frontal debe ser proporcional a la velocidad de la unidad, ya que a mayor velocidad más se avanza sin control mientras el supervisor evalúa la situación, así como la frenada puede ser más larga cuando finalmente se decide pausar la unidad. Como además del Supervisor, los vehículos van equipados con un LiDAR como medida última anti-colisiones, los márgenes también deben aumentar la superficie de detección lo suficiente para que sea siempre mayor que el rango de detección del láser, ya que en caso contrario podría saltar el LiDAR primero y se trata de un sistema que para totalmente el vehículo de forma que sólo puede ser arrancado manualmente.

La Figura 4.12 muestra la superficie de detección de una unidad en función de las dimensiones reales del apero, el vehículo, la pista y el rango del LiDAR. La ecuación (4.1) indica el rango frontal que usaba el LiDAR, siendo v la velocidad de la unidad y α el ángulo de inclinación del laser. Este mismo rango se usó para definir el margen frontal de la superficie de detección del *supervisor de colisiones*.

$$\text{MargenFrontal}(v) = (1,89 \cdot v + 1) \cdot \cos \alpha \quad (4.1)$$

Por último, el margen trasero es menos crítico, porque nunca o rara vez se circula marcha atrás, y basta con fijarlo con una distancia de alrededor de un metro.

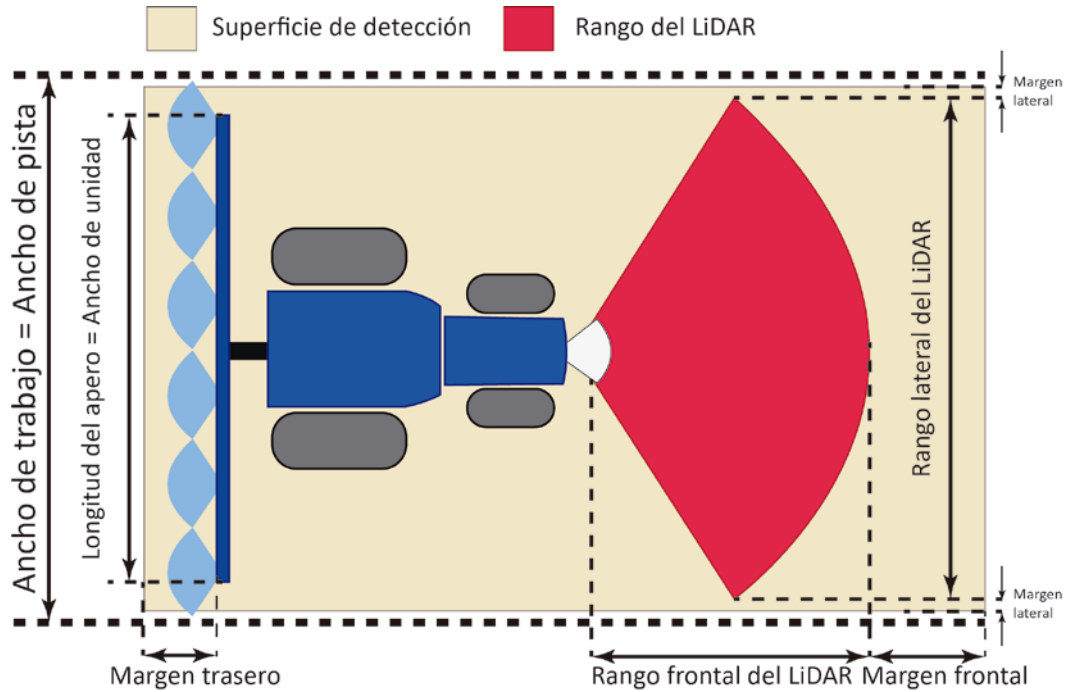


Figura 4.12. Superficie que estima el *supervisor de colisiones* que ocupa una unidad

Las superficies se avanzan hasta cierto máximo tiempo u horizonte futuro para no sobrecargar el sistema detectando colisiones muy lejanas que quizás ni se produzcan por las irregularidades del terreno y la variabilidad en los tiempos de respuesta de las SCUs. El valor horizonte debe definirse de tal modo que deje suficiente tiempo de reacción al operario para evaluar la situación y tomar alguna medida, como por ejemplo parar alguna de las unidades. Además del propio código de alarma, la notificación incluye información extra como el riesgo de colisión, calculado a partir del tiempo que queda para que se produzca, así como la posición en la que se calcula que tendrá lugar el choque.

Puesto que las situaciones detectadas por este supervisor son situaciones críticas y potencialmente muy peligrosas, es interesante enlazar directamente las alarmas a alguna actuación de neutralización para contrarrestar los daños lo antes posible. Una potencial actuación es pausar una de las unidades para dejar vía libre a la otra evitando así la colisión, aún así existen casos donde ambas unidades deben pausarse, por ejemplo, cuando se aproximan siguiendo la misma trayectoria pero en sentidos distintos. Como se explicó con anterioridad, el Módulo de Reparación de Fallos es el encargado de ordenar estas actuaciones, asociando alarmas y órdenes. Ahora bien, además de pausar las unidades es importante reanudar su movimiento cuando el riesgo desaparece, para ello es necesario detectar cuándo las unidades tienen de nuevo vía libre para continuar con la misión. Puesto que el *supervisor de colisiones* está dedicado a detectar intersecciones y solapamientos que implican colisiones, puede también detectar cuando el camino está

libre, por lo que se añadió a este supervisor un tercer caso de funcionamiento, en concreto, un caso que comprueba si hay alguna unidad pausada, y en caso afirmativo consulta si se chocaría con otra en caso de reanudar su movimiento. El procedimiento es similar, se avanza su posición y las de las demás incrementalmente y si no existe ningún solapamiento, entonces no hay potencial colisión y la unidad puede ser reanudar. Para notificar esta situación, el supervisor emite una alarma de camino libre.

El *supervisor de misión completada* recibe directamente la información generada por los supervisores de unidad del nivel inmediatamente inferior y analiza las alarmas de misión restante para comprobar si se ha realizado el 100 % de la misión en alguna de las unidades. En caso de ser así lo registra y cuando todas han terminado emite la alarma de fin de misión.

Por último, el *supervisor de flota* es el encargado de concentrar todas las alarmas de la flota y de reenviárselas al Módulo de Reparación de Fallos y al Gestor de Notificación de Alarmas.

Todos los supervisores del Supervisor de Misión se desarrollaron en C++ usando las librerías Qt (Qt Libraries, 2015), con el objetivo de obtener tiempos de respuesta adecuados a la gran cantidad de información a monitorizar y analizar. Las librerías Qt simplificaron la implementación de determinados aspectos técnicos, como la detección de las colisiones, usando las clases *QtGraphicsScene*, las comunicaciones internas para propagar las alarmas entre los diferentes módulos, usando la metodología de *signals/slots*, y las comunicaciones con módulos externos, mediante los sockets del paquete de clases comunicaciones de Qt.

Las contribuciones y resultados del sistema Supervisor propuesto formado por el Supervisor de Misión descrito y por el Módulo de Reparación de Fallos y el Gestor de Notificación de Fallos que se expondrán a continuación se publicaron en (Conesa-Muñoz, Gonzalez-de-Soto, Gonzalez-de-Santos, & Ribeiro, 2015).

4.3.2. Implementación del Módulo de Reparación de Fallos

Como se ha comentado, el Módulo de Reparación de Fallos puede enviar órdenes a las SCUs cuando recibe ciertas alarmas preestablecidas. En la implementación que se realizó para el proyecto RHEA, solamente las alarmas generadas por los supervisores de más alto nivel, el de colisiones y el de misión completa, desencadenan actuaciones. La Tabla 4.6 muestra las actuaciones que se asociaron a cada alarma.

Tabla 4.6. Relaciones alarma-actuación del Módulo de Reparación de Fallos.

Alarma	Actuación
Muy cerca	Parar unidades
Colisión	Pausar uno o dos unidades
Camino libre	Reanudar unidad
Misión completada	Parar todas las unidades de la flota

Cuando las unidades están muy cerca, directamente el Módulo de Reparación las para por completo (orden *Stop* de la Tabla 4.3) para prevenir cualquier contacto. Si se predice una colisión próxima, la situación no es tan peligrosa como en el primer caso (hay más tiempo de reacción) y de ahí que se envíe una orden *Pausa* (véase Tabla 4.3) para

que, una vez que el camino está libre, se puede reanudar la unidad, mediante una orden *Reanudación* (véase Tabla 4.3).

Por último, si se recibe una alarma de misión completada, se envía una orden de parada (*Stop*) a todas las unidades para que definitivamente paren de funcionar todos los dispositivos menos las comunicaciones.

Al igual que el Supervisor de Misión, el Módulo de Reparación de Fallos se desarrolló usando Qt para poder asociar eficientemente las alarmas y las acciones mediante la metodología *signals/slots*.

4.3.3. Implementación del Gestor de Notificación de Alarmas

En el sistema que se implementó para RHEA, todas las alarmas generadas se retransmitieron hasta el operario, excepto las de vía libre y parte de las de velocidad, ya que, como se verá más adelante, las unidades tenían dificultades para mantener el ritmo indicado y se generaban constantemente. Por ello, se limitó su emisión a una por segundo, para no sobrecargar el sistema ni saturar al operario.

4.4. Resultados

Se planificaron dos misiones para probar el sistema de supervisión mediante las aproximaciones propuestas en este trabajo de investigación y descritas en el Capítulo 3. La primera fue diseñada para probar el sistema con un tractor llevando a cabo un tratamiento de pulverización localizada de malas hierbas a partir de un mapa que indicaba la distribución de los rodales en un cultivo. La segunda misión se diseñó para probar la supervisión de la flota, es decir, con varias unidades trabajando a la vez y así evaluar el funcionamiento del *supervisor de colisiones* y del Módulo de Reparación de Fallos. En esta última misión dos vehículos tenían que recorrer todo el campo coordinadamente simulando llevar a cabo un tratamiento entre los dos.

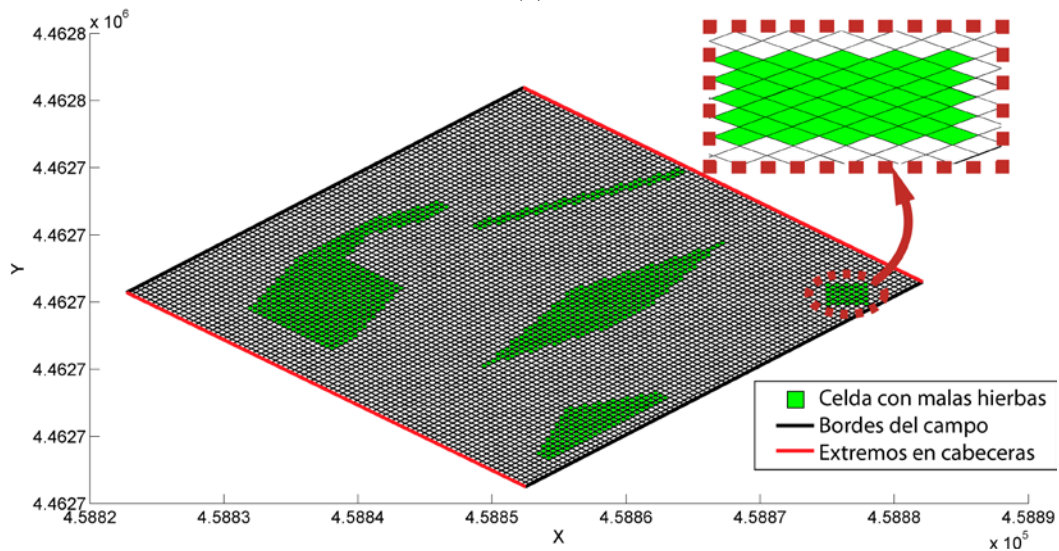
Además de validar la supervisión, el objetivo de estas pruebas era corroborar que las planificaciones generadas por el sistema Planificador del Capítulo 3 son válidas y pueden emplearse con éxito con tractores de una flota real.

Las misiones se ejecutaron en uno de los campos del Centro de Automática y Robótica en Madrid [40°18'51.102"N, 3°29'03.379"W]. El campo era de dimensiones 42,5 m × 41,5 m y se dibujaron distintos contornos con yeso que simulaban los rodales de malas hierbas (véase Figura 4.13a). Asimismo, se construyó un mapa de malas hierbas a partir de geo-referenciar manualmente con un GPS-RTK los contornos de los rodales dibujados. El campo se representó en una matriz a partir de la descomposición del mismo en una cuadrícula donde cada celda equivalía a una superficie de 0,5 m x 0,5 m del cultivo, coincidiendo con el ancho de pulverización de cada boquilla (consúltese la barra de pulverización de la sección 4.2.2). Cada componente de la matriz se puso a 1 cuando la celda que representaba estaba dentro de alguno de los contornos dibujados (zona infestada) y a 0 en caso contrario. Tras el proceso descrito, el mapa o matriz de distribución de malas hierbas resultante para el campo considerado tenía el aspecto que se muestra en la Figura 4.13b. Los SCUs se configuraron para enviar mensajes de monitorización con una frecuencia de 250 ms (4 Hz) para ambas misiones. Esta frecuencia se consideró suficiente porque en ambas pruebas se fijó la velocidad máxima

de los vehículos en 3 km/h. A esta velocidad se avanza a 0,83 m/s, es decir, 21 cm por mensaje. Se empleó esta velocidad, porque era la que garantizaba, por pruebas realizadas con anterioridad y que se pueden consultar en el trabajo (Emmi, 2014), que las unidades fueran capaces de navegar autónomamente con estabilidad. Las misiones se ejecutaron varias veces y los resultados fueron similares en todos los ensayos. A continuación se describen los resultados obtenidos y que fueron publicados en (Conesa-Muñoz, Gonzalez-de-Soto, Gonzalez-de-Santos, & Ribeiro, 2015).



(a)



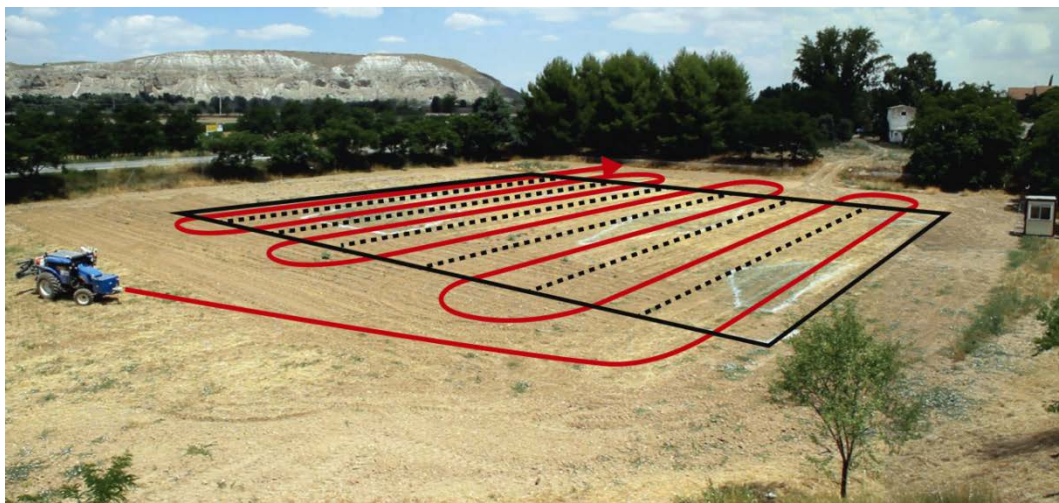
(b)

Figura 4.13. (a) Campo con los rodales dibujados con líneas blancas. (b) Representación matricial del campo; mapa de distribución de malas hierbas.

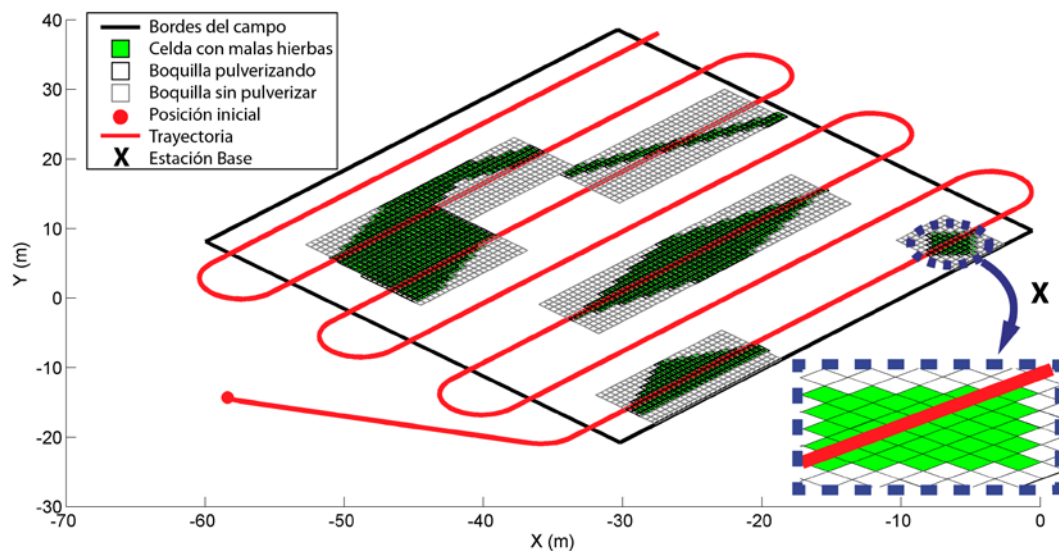
4.4.1. Test de misión de pulverización localizada de malas hierbas

En esta misión una unidad equipada con la barra de pulverización explicada en la sección 4.2.2 tenía que aplicar el herbicida solo en las posiciones indicadas en el mapa de malas hierbas. En (Conesa-Muñoz & Ribeiro, Spraying Mission Test, 2015) puede reproducirse un vídeo donde se ve la misión completa.

Como el ancho de trabajo de la barra es 6 m, el campo se dividió en 7 calles, y se calculó mediante el Planificador presentado en el Capítulo 3 la trayectoria y el plan de actuación que se muestra en la Figura 4.14, además, a modo de ejemplo, se ofrece un fragmento del plan en la Tabla 4.7, en concreto, se muestran los primeros pasos del plan, hasta la finalización del primer rodal.



(a)



(b)

Figura 4.14. Trayectoria y estado de las boquillas durante la misión.

Tabla 4.7. Plan de actuación del vehículo hasta la finalización del primer rodal (véase trayectoria de la Figura 4.14).

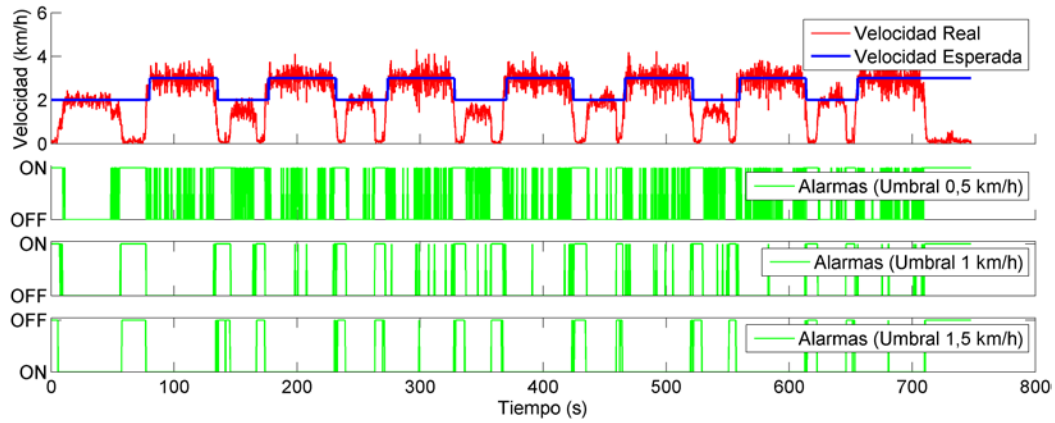
	Posición	Orient.	Vel. (km/h)	Estado de las boquillas de la barra de pulverización (0 = OFF, 1 = ON)											
				V1	V2	V3	V4	V5	V6	V7	V8	V9	V10	V11	V12
1	(-32.40, -18.70)	45,85	2	0	0	0	0	0	0	0	0	0	0	0	0
2	(-29.61, -15.83)	45,85	3	0	0	0	0	0	0	1	1	1	0	0	0
3	(-28.92, -15.12)	45,85	3	0	0	0	0	0	1	1	1	1	0	0	0
4	(-28.22, -14.40)	45,85	3	0	0	0	0	1	1	1	1	1	0	0	0
5	(-27.52, -13.68)	45,85	3	0	0	0	1	1	1	1	1	1	0	0	0
6	(-26.83, -12.96)	45,85	3	0	0	1	1	1	1	1	1	1	0	0	0
7	(-26.13, -12.25)	45,85	3	0	1	1	1	1	1	1	1	1	0	0	0
8	(-25.44, -11.53)	45,85	3	1	1	1	1	1	1	1	1	1	0	0	0
9	(-24.74, -10.81)	45,85	3	0	1	1	1	1	1	1	1	1	0	0	0
10	(-24.04, -10.09)	45,85	3	0	0	1	1	1	1	1	1	1	0	0	0
11	(-23.35, -9.38)	45,85	3	0	0	0	1	1	1	1	1	1	0	0	0
12	(-22.65, -8.66)	45,85	3	0	0	0	0	1	1	1	1	1	0	0	0
13	(-21.95, -7.94)	45,85	3	0	0	0	0	0	1	1	1	1	0	0	0
14	(-21.26, -7.22)	45,85	3	0	0	0	0	0	0	1	1	1	0	0	0
15	(-20.56, -6.51)	45,85	3	0	0	0	0	0	0	0	0	0	0	0	0
...

Las celdas resaltadas en la Figura 4.14 delimitan las posiciones en las que la barra debe estar activa y cada celda representa la superficie de tratamiento que abarca una boquilla. Si el contorno de la celda es gris significa que la boquilla que cubre esa zona debe estar cerrada y si es negra, abierta.

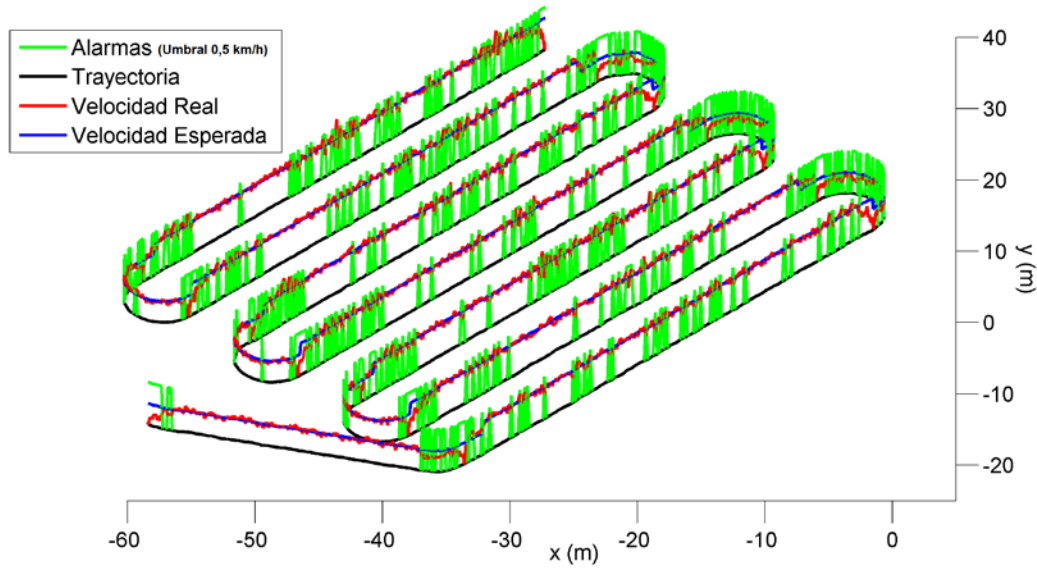
La Figura 4.15 muestra los resultados obtenidos por el *supervisor de velocidad*. La velocidad planificada era 3 km/h dentro del cultivo y 2 km/h en cabeceras, por la mayor complejidad de los giros. Se probaron tres márgenes de error: 0,5 km/h, 1 km/h y 1,5 km/h. Las velocidades reales y esperadas se muestran respectivamente en rojo y azul en la parte superior de la figura, y las alarmas en la parte inferior. Como se observa, la unidad tuvo serias dificultades para mantener las velocidades planificadas, especialmente en las transiciones entre pistas, donde el cambio de trayectoria y velocidad se llevaba a cabo de forma muy lenta (las unidades llegaban a pararse). Por esta razón el umbral de 0,5 km/h era muy ajustado y se produjeron muchas alarmas (véase Figura 4.15), por lo que se decidió aumentarlo para comprobar si el supervisor generaba un número inferior de alarmas, así sucedió con umbrales de 1 km/h y 1,5 km/h, tal como se puede ver en la Figura 4.15a. En definitiva se puede concluir que el *supervisor de velocidad* funcionó correctamente.

La Figura 4.16a muestra los resultados para el *supervisor del estado del apero*, en este caso la barra de pulverización, para la misión completa mientras que la Figura 4.16b muestra solo los resultados en el intervalo transcurrido entre el segundo 285 y el 321. Los estados reales y esperados de las 12 boquillas aparecen en rojo y azul respectivamente y la activación/desactivación de las alarmas en verde. Las diferencias entre los estados se deben principalmente al tiempo de respuesta de las válvulas de las boquillas que se detectaron satisfactoriamente. Como el supervisor comprueba el estado del apero con una frecuencia superior, cada 250 ms, al tiempo de respuesta de las válvulas, que es del orden de 1 a 2 s, cada vez que se ordena abrir una boquilla el supervisor detecta una inconsistencia en el estado hasta que la válvula responde a la orden. Como las diferencias duran muy poco tiempo, del orden de segundos, se ha ampliado el intervalo en la Figura 4.16b para poder distinguir con facilidad.

Es interesante observar que los estados de las boquillas encajan con precisión en la forma que tiene los rodales considerados, por ejemplo, el comportamiento de apertura/cierre de las boquillas en el rodal con forma de rombo de la tercera pista se puede distinguir claramente en torno al segundo 300.



(a)



(b)

Figura 4.15. Alarmas generadas por el *supervisor de la velocidad* en función de la velocidad real, la esperada y el margen (umbral) de error utilizado. (a) Alarmas generadas a lo largo del tiempo para umbrales de 0,5 km/h, 1 km/h y 1,5 km/h. (b) Alarmas generadas a lo largo de la trayectoria para un umbral de error de 0,5 km/h

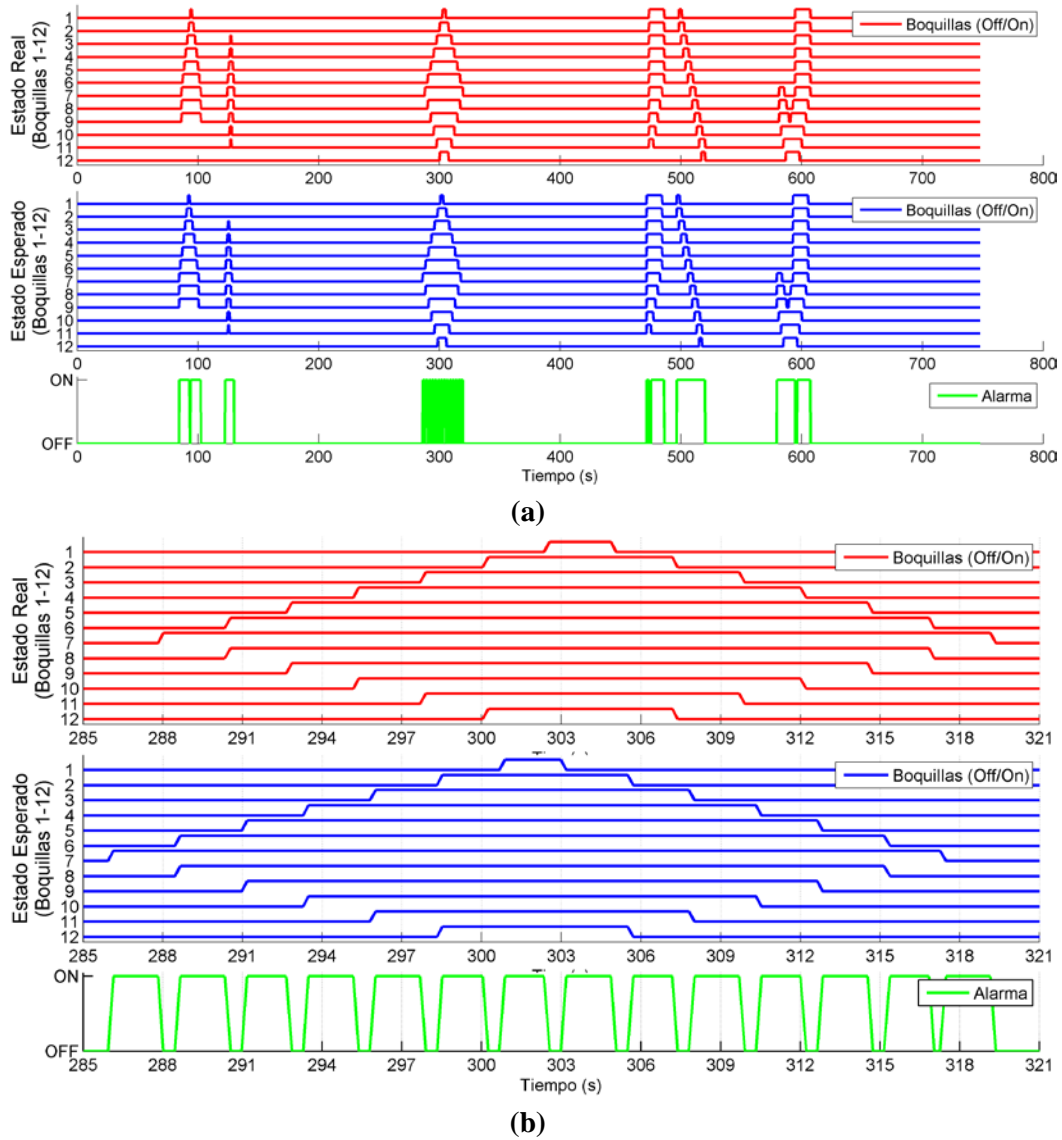


Figura 4.16. Alarmas generadas por el *supervisor de estado del apero* en función del estado real y el esperado durante (a) la misión completa o recorrido completo del campo y (b) en el intervalo que transcurre entre los segundos 285 y 321.

La Figura 4.17 muestra los resultados obtenidos por el *supervisor de fuera de trayectoria*. Este supervisor considera un punto como visitado cuando la distancia entre la posición actual y la esperada es inferior a 10 cm. La trayectoria real y la esperada aparecen en rojo y azul respectivamente y las activaciones/desactivaciones de las alarmas en verde. Como se aprecia, la mayoría de las alarmas se generaron durante las maniobras de giro, donde los vehículos se movían con menos precisión debido a las condiciones del terreno que provocaban que las ruedas patinasen, ya que había llovido unos días antes, y porque el radio mínimo de giro del vehículo era de 2,89 m y la misión se planificó

considerando un radio de 3 m para comprobar el funcionamiento del supervisor cuando la maquinaria funciona en condiciones extremas.

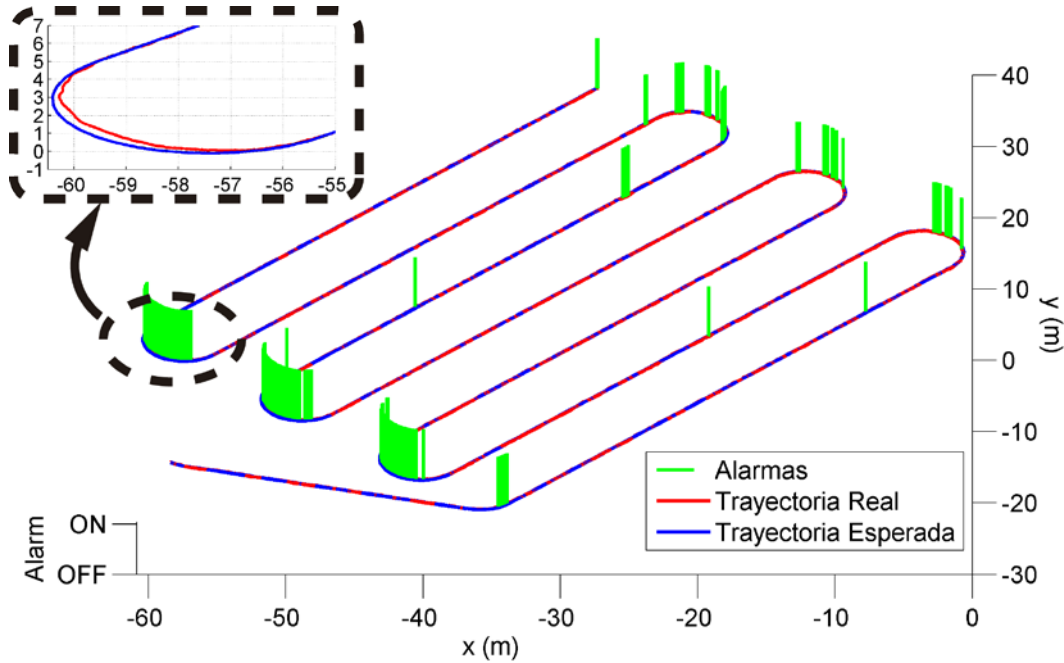


Figura 4.17. Alarmas generadas por el *supervisor de trayectoria* en función de la trayectoria real y la esperada

La Figura 4.18 muestra los resultados del *supervisor de misión restante*. Como la unidad siguió la ruta prácticamente sin desviarse, en todo momento se pudo calcular con bastante precisión el porcentaje de trayectoria restante a partir de la posición actual. Puesto que la velocidad durante los giros era inferior (2 km/h) a la de las rectas (3 km/h), el porcentaje decrecía menos durante las maniobras, de ahí que la pendiente de la Figura 4.18 no sea constante.

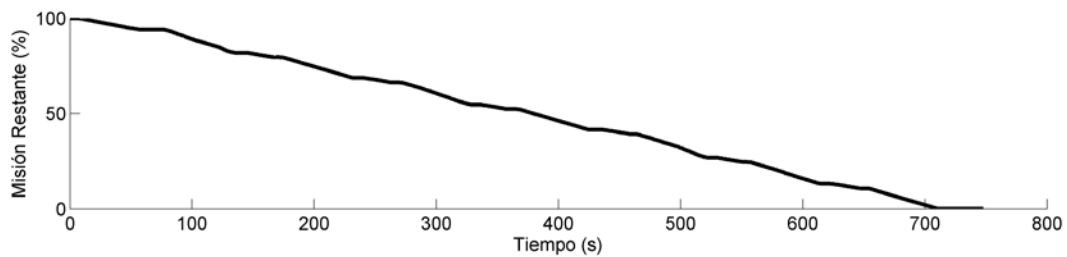


Figura 4.18. Información generada por el *supervisor de misión restante*.

La Figura 4.19 muestra los resultados del *supervisor de interrupción*. Este supervisor se configuró para monitorizar las localizaciones proporcionadas por el GPS del vehículo por ser información crítica. Se probaron dos umbrales de tiempo máximo sin recibir nada: 0,5 s y 1 s. Como la frecuencia de emisión desde las SCUs se había fijado en 4 Hz, la

mayoría de los mensajes se recibieron con una separación media de 250 ms, como se muestra en la figura. Sin embargo, en un caso la localización no se recibió durante un tiempo cercano a los 0,6 s, probablemente debido a algún retraso en la red de comunicaciones. Como este intervalo no superó el segundo, no se generó ninguna alarma para el umbral de 1 s por lo que el comportamiento de las alarmas para este umbral no se muestra en la Figura 4.19. En el caso del umbral de 0,5 s sí se generó una alarma como se puede ver alrededor del segundo 300.

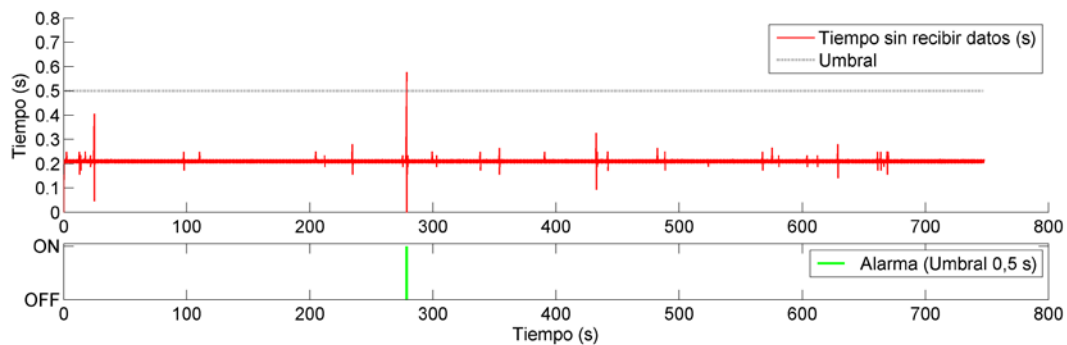


Figura 4.19. Alarmas generadas por el *supervisor de interrupción de servicios*.

Por último, las Figura 4.20 y Figura 4.21 muestran las alarmas emitidas por el *supervisor de unidad*, el encargado de concentrar todas las alarmas relacionadas con un mismo vehículo. Las alarmas de velocidad visualizadas son las del margen de error 0,5 km/h. Para visualizar la alarma por interrupción se usó el umbral 0,5 s.

Para esta misión no se muestran los supervisores del nivel de flota porque el experimento se realizó con un único vehículo y por tanto estos supervisores no aportan ninguna información adicional, es más algunos ellos ni se activaron como por ejemplo, por razones obvias, el de colisiones.

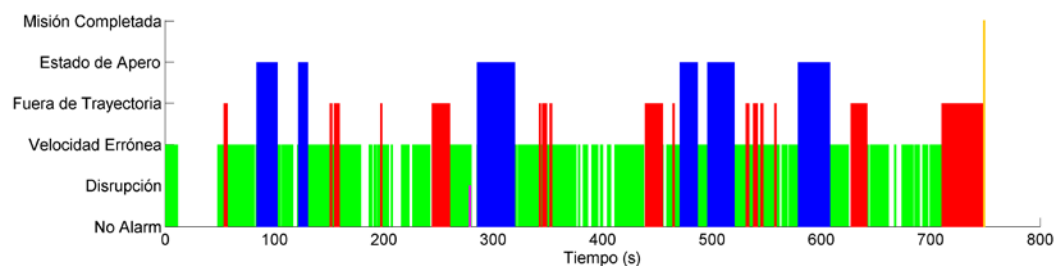


Figura 4.20. Alarmas generadas por el *supervisor de unidad* a lo largo del tiempo

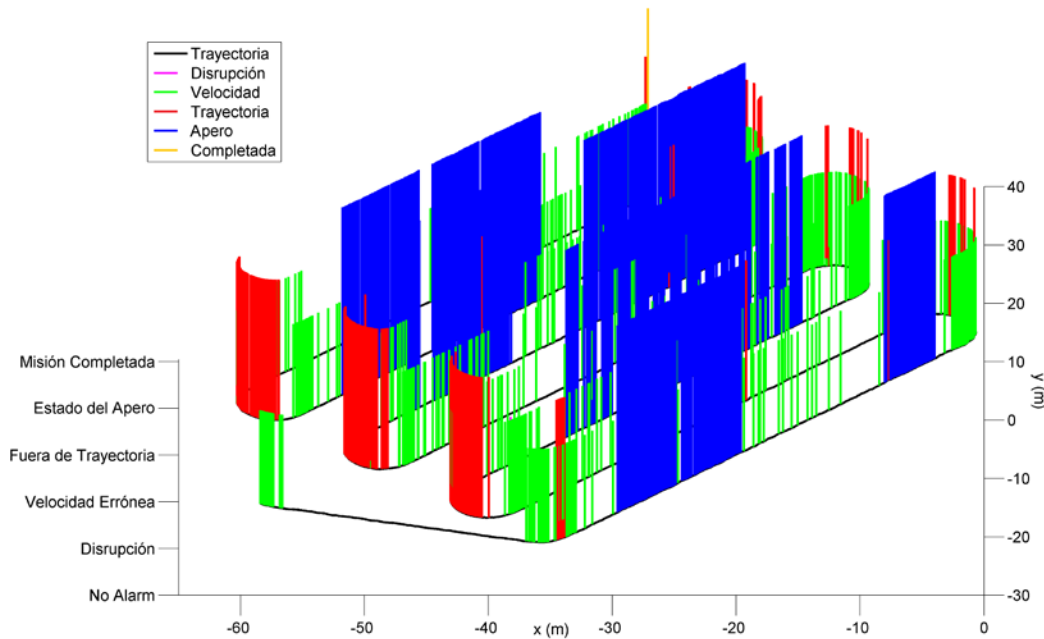


Figura 4.21. Alarmas generadas por el *supervisor de unidad* a lo largo de la trayectoria.

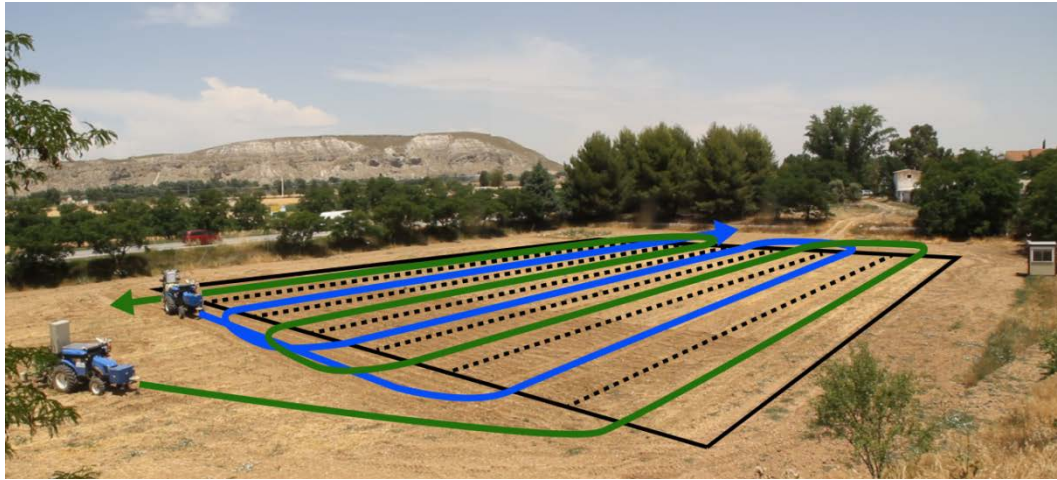
4.4.2. Test de colisiones

La segunda misión se diseñó para verificar el sistema Supervisor desarrollado con varias unidades trabajando a la vez y así evaluar los supervisores de flota, especialmente el de colisión. La misión fue grabada y se puede reproducir en (Conesa-Muñoz & Ribeiro, Collision Supervisor Test, 2015).

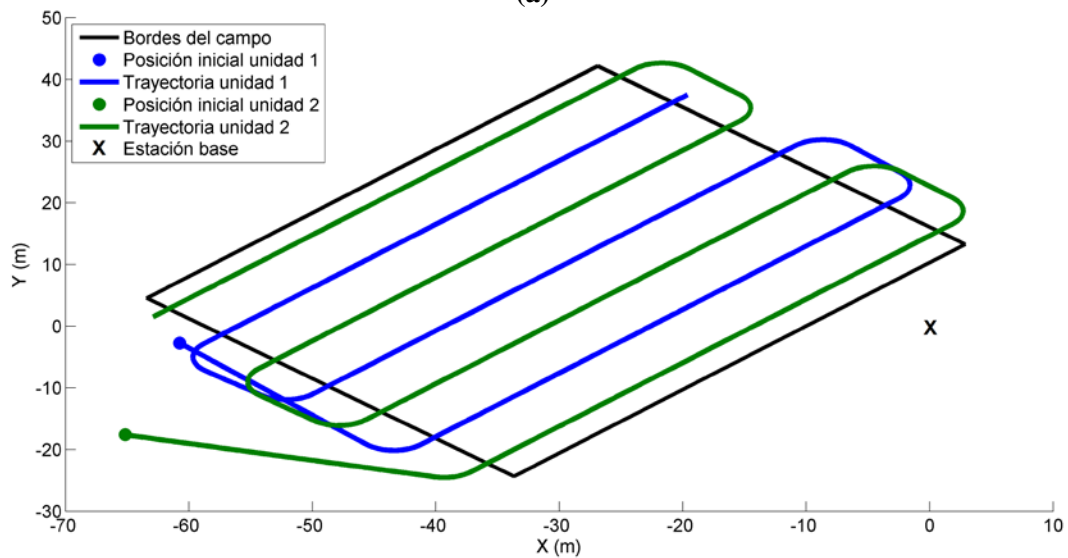
Los tractores tenían que cubrir el campo entero siguiendo trayectorias paralelas, a la misma velocidad y simulando un tratamiento de pulverización, por tanto el Supervisor supuso que las unidades estaban trabajando con barras de 6 m (consúltese sección 4.2.2). Las trayectorias estaban pensadas para que las unidades tuvieran que girar en intervalos cercanos, en posiciones relativamente próximas para así propiciar la detección de potenciales colisiones en las cabeceras, como se aprecia en las Figuras 4.22 y 4.23.

La Figura 4.22 muestra las posiciones iniciales, los contornos del campo (líneas negras) y las trayectorias, línea azul para el vehículo 1 y verde para el vehículo 2. La Figura 4.23 muestra los momentos clave durante la misión, concretamente aquellos en los que el *supervisor de colisiones* detecta un potencial choque entre unidades o vía libre para reanudar la marcha de una unidad previamente pausada.

La misión comienza con ambos tractores moviéndose, pero enseguida se detecta una colisión cerca de la esquina inferior derecha del campo (Figura 4.23a), en donde ambos vehículos tienen que girar relativamente cerca. Hay que contar con que los vehículos simulan llevar barras de 6 metros y además hay unos márgenes de seguridad con los que trabaja el supervisor.



(a)



(b)

Figura 4.22. Trayectorias de los dos vehículos en la segunda misión.

Una vez detectada la colisión (cruz roja en Figura 4.23a), la alarma llega hasta el Módulo de Reparación de Fallos que ordena pausar la unidad 1 (trayectoria azul). Mientras tanto la unidad 2 (trayectoria verde) continúa moviéndose y cuando está a suficiente distancia, el supervisor detecta que la primera unidad tendría el camino libre de colisiones en caso de que reanudara la marcha. Automáticamente genera una alarma de vía libre que llega hasta el Módulo de Reparación de Fallos, que ordena a la unidad que reanude la marcha (Figura 4.23b). Como las trayectorias son paralelas y las unidades deben moverse durante toda la misión a las mismas velocidades según lo especificado en el plan de actuación, es decir, a 3 km/h en cultivo y a 2 km/h en cabeceras, y como, además, la segunda unidad ha adquirido cierta ventaja respecto a la primera, podría esperarse que los vehículos no volvieran a coincidir en posiciones cercanas en toda la misión.

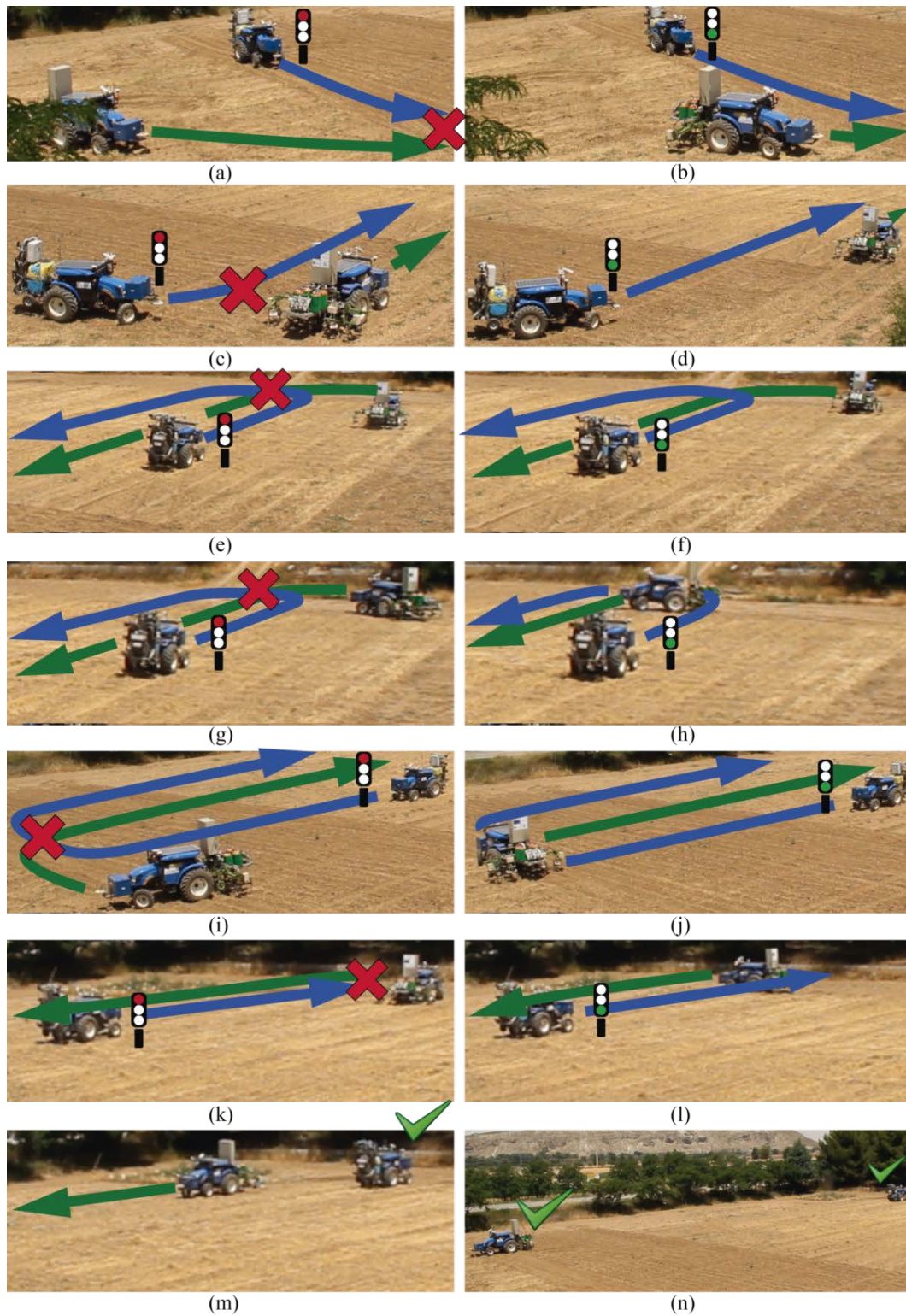


Figura 4.23. Cronología de la misión: colisiones resueltas mediante pausas (a, c, e, g, i, k), reanudaciones de la marcha (b, d, f, h, j, l) y puntos finales (m, n).

Sin embargo, debido a razones mecánicas, la segunda unidad se movía más lentamente que la primera, de hecho en el vídeo se aprecia claramente que tarda más en girar y arrancar al comienzo de cada pista, de ahí que a la primera unidad tenga tiempo a alcanzar una nueva posición peligrosa que de nuevo es detectada por el *supervisor de colisiones* (Figura 4.23c) y que de nuevo el Módulo de Reparación de Fallos tenga que dar la orden de pausar la unidad 1 para prevenir la potencial colisión. Aunque en el vídeo parezca que hay espacio suficiente para que la unidad primera avance un poco más sin chocar, hay que recordar que las superficies que utiliza el supervisor son considerablemente más anchas que los propios tractores, en concreto determinadas por el ancho de la barra de tratamiento más un margen de seguridad. Finalmente la segunda unidad acaba el giro y entra en la primera pista y se puede reanudar la marcha de la primera unidad (véase Figura 4.23d). Durante el segundo giro de la unidad 2, (véanse Figura 4.23e y g) sucede lo mismo que anteriormente, es decir, la unidad tarda más tiempo del esperado en tomar la curva y la primera unidad tiene que ser pausada dos veces. Al igual que antes, como el *supervisor de colisiones* calcula las posiciones futuras a partir de las velocidades planificadas, predice que ya hay espacio suficiente para reanudar la primera unidad y emite la alarma de camino libre, pero tiene que volver a notificar la colisión cuando ve que la segunda unidad se vuelve a retrasar. En las siguientes dos maniobras de giro, se detectaron de nuevo dos colisiones por las mismas razones expuestas (véanse Figura 4.23i y k), por lo que hubo que pausar y reanudar la unidad 2 otras dos veces (véanse Figura 4.23h y j). Tras esto, ambas unidades terminaron satisfactoriamente los subplanes de actuación encomendados (véanse Figura 4.23m y n).

La Figura 4.24 muestra las salidas que emitió el *supervisor de colisiones* y la evolución de los estados internos de las unidades que usó para poder controlar apropiadamente la misión. Las alarmas de colisión y camino libre emitidas aparecen en rojo y verde respectivamente, y, puesto que se detectaron y resolvieron seis colisiones, hay seis líneas rojas (véanse Figura 4.23a, c, e, g, i, k) y seis líneas verdes (véanse Figura 4.23b, d, f, h, l).

La unidad 2 mantuvo el mismo estado durante toda la misión porque el Módulo de Reparación de Fallos decidió que fuera siempre el vehículo 1 el que se pausara/reanudara. En la primera colisión el Módulo de Reparación de Fallos eligió aleatoriamente pausar esa unidad, ya que en ese momento cualquiera de las dos posibilidades era válida e igual de ventajosa, y con esta decisión determinó que en la siguientes colisiones la única opción posible fuera pausar la primera unidad, ya que pausar la segunda unidad no habría resuelto nada al ir siempre por delante.

Con cada colisión detectada, el supervisor registra la unidad en el estado *Pausando (Por Petición de Supervisor)* y la mantiene ahí hasta que la orden se procesa, es decir, el supervisor se entera porque le llega el estado de la unidad a través de la información de monitorización emitida por la SCU, entonces el supervisor cambia el estado de la unidad a *Pausado (Por Petición de Supervisor) – Etapa 1*, en el que permanece hasta que se detecta que el camino está libre de colisiones y se pasa a la segunda etapa de la pausa cambiando el estado a: *Pausado (Por Petición de Supervisor) – Etapa 2*. En este estado el vehículo tiene que esperar 10s antes de reanudar definitivamente la marcha extremando de este modo las medidas de seguridad de la flota al comprobar en los experimentos que en muchos casos las unidades no respondían a tiempo a las órdenes recibidas. Por tanto, se decidió que era mejor darles a las unidades un margen extra para que tuvieran tiempo de abandonar las posiciones conflictivas y, así, no detectar de nuevo la misma colisión, como sucede en las colisiones de las Figura 4.23c y e. El margen de 10 s se determinó

experimentalmente por un procedimiento de prueba y error. Tras ese tiempo se pasa al estado *Reanudando* y después al estado *En movimiento*, cuando el supervisor, a partir de la información de monitorización, detecta que la unidad ya está en marcha.

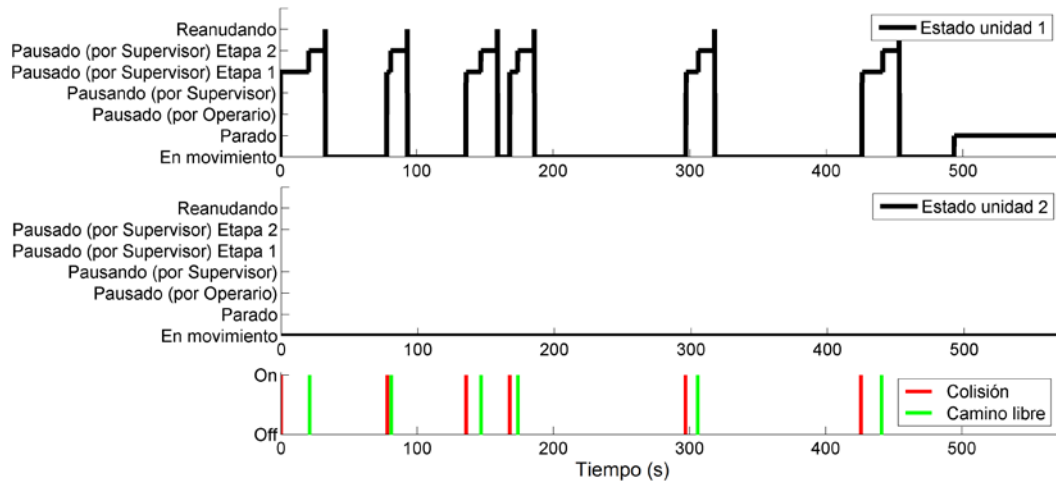


Figura 4.24. Alarmas generadas y evolución de los estados internos que el *supervisor de colisiones* usa durante la misión.

Por último, la Figura 4.25 muestra los resultados del *supervisor de misión completada*. Este supervisor recibe como entradas las alarmas generadas por los *supervisores de misión restante* de cada unidad. Como ambas unidades completaron la misión (líneas en rojo en la figura), tan pronto como terminó la última de las unidades se consideró la misión global completada (línea verde).

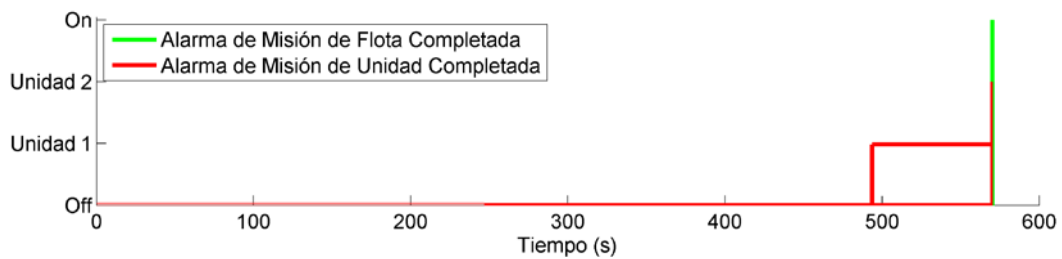


Figura 4.25. Salidas y entradas del *supervisor de misión completada*.

5 GESTOR DE MISIONES

Mediante la planificación descrita en el Capítulo 3 y la supervisión del Capítulo 4 es posible planificar una misión y monitorizar su ejecución. Ambas partes son pasos consecutivos necesarios para la automatización total de multitud de escenarios agrícolas, como por ejemplo los considerados en el proyecto europeo RHEA y descritos en el capítulo de introducción (sección 1.3). En el caso del proyecto RHEA, además, se abordaba la inspección previa al tratamiento, mediante una flota de vehículos aéreos no tripulados o drones. El proyecto integraba una tarea de inspección, realizada por las unidades aéreas de la flota, de la que se obtenía información estructurada en forma de mapas que posteriormente era utilizada en la tarea de tratamiento, realizada por las unidades terrestres de la flota. Resultó por tanto esencial integrar todos los módulos y gestionar adecuadamente el flujo de trabajo de todo el sistema RHEA con objeto de automatizar al máximo la ejecución de la misión, simplificando la labor del operario. Ésta es la principal motivación del diseño y desarrollo del sistema tratado en este capítulo, el Gestor de Misiones.

El Gestor de Misiones planteado se describe como una propuesta general de integración de robótica aérea y terrestre en aquellos escenarios, frecuentes en problemas medioambientales, que se benefician de una etapa de vigilancia y valoración de daños, previa a la etapa de intervención dirigida a paliar el problema. En estos casos, la vigilancia/valoración se puede realizar con vehículos aéreos sobre todo cuando las zonas implicadas son muy extensas, y la intervención con vehículos terrestres ya que en la mayoría de casos la solución del problema requiere la acción localizada con equipo específico y pesado; este es el caso, por ejemplo, de las unidades terrestres para tratamiento con fitosanitarios.

La primera sección explica la estructura general que se propone para el Gestor y los diferentes módulos que la componen. La segunda explica la implementación en la que finalmente se concretó este sistema para RHEA. Y, por último, la sección final describe las pruebas que se llevaron a cabo para evaluar todo el sistema implementado y los resultados obtenidos.

5.1. Estructura

El Gestor de Misiones que se propone debe estar compuesto internamente por los módulos o sistemas independientes necesarios para automatizar escenarios tan complejos como aquellos que combinan misiones aéreas de inspección y terrestres de tratamiento.

La Figura 5.1 muestra la estructura interna que se propone para el Gestor de Misiones y los sistemas externos con los que puede interactuar.



Figura 5.1. Estructura interna del Gestor de Misiones planteado y sistemas externos con los que puede interactuar.

Los módulos que componen el Gestor de Misiones son los siguientes:

- Un planificador de misión aérea y uno de misión terrestre para generar los planes que las unidades tienen que seguir para completar las misiones. Ambos módulos no pueden unirse en un único planificador por las diferencias inherentes entre los vehículos aéreos y terrestres, además de las características diferentes de las misiones (vigilancia/valoración frente a tratamiento/intervención).
- Un controlador de misión aérea y otro de misión terrestre para automatizar las misiones a nivel de flota. Aunque las unidades, consideradas por separado, son autónomas, la flota tiene que ser coordinada, por ejemplo para lanzar/pausar/reanudar/parar la misión en todos los vehículos simultáneamente. Además, estos controladores también deben encargarse de codificar los planes con las operaciones exactas soportadas por las unidades y de transmitirlos a los distintos robots.
- Un supervisor de misión aérea y otro de misión terrestre para monitorizar y comprobar que las misiones se ejecutan acorde a los planes generados y, en caso de no ser así, notificarlo mediante alarmas.
- Un sistema para procesar los datos obtenidos durante las misiones, especialmente las de inspección, y poder utilizarlos en misiones posteriores. Por ejemplo, en el caso de

que mediante una misión se hubieran adquirido imágenes de un cultivo, incluiría un sistema capaz de analizar las texturas y colores de las imágenes y determinar la posición de elementos de interés como las líneas de cultivo y los rodales de malas hierbas.

- Por último, un despachador para gestionar el flujo de trabajo necesario para completar todo el proceso. Para ello, el despachador encapsula las conexiones a todos los módulos incluidos en el Gestor de Misiones y vuelve a dirigir el proceso a los módulos apropiados cuando es necesario. Aún más, recoge, procesa y redirige las consultas (planes, ejecuciones, pausas, reanudaciones, etc.) de los sistemas externos (GUI) si el operario quiere controlar activamente el flujo de trabajo. Este módulo es especialmente importante, ya que permite la conexión de nuevos módulos al Gestor de Misiones ampliando, así, las funcionalidades del mismo.

Además de los subsistemas internos, es importante conocer los externos que pueden comunicarse con el Gestor de Misiones. Estos sistemas, que también se muestran en la Figura 5.1, son los siguientes:

- Una interfaz gráfica de usuario (en inglés *Graphical User Interface*, GUI) para que el operario pueda realizar peticiones al Gestor de Misiones tales como planificaciones, ejecución de misiones, mosaicos, tele-operación de alguna unidad, etc., y para visualizar la información desencadenada por las peticiones. En siguientes apartados se muestran las peticiones que se pueden realizar y los procesos que desencadenan.
- Una GUI portátil para ser utilizada en determinadas situaciones, tales como averías, revisión del equipo, etc., que requieren la presencia de un operario en campo. En esos casos el operario debe poder desplazarse por el campo sin dejar de recibir la información de la misión y, por supuesto, pudiendo tomar el control de algún elemento de la flota si esto es necesario y siempre previa petición al Gestor de Misiones que se estará ejecutando en una estación base donde además estará otro operario que supervisa toda la misión desde la GUI principal.
- Los drones y los vehículos terrestres, para llevar a cabo las misiones.
- Una base de datos (BD) donde se almacena información de interés para las misiones, tal como los contornos de los campos, las características de las unidades o incluso los datos adquiridos en misiones anteriores, etc., así como los datos que se generan durante la ejecución de la misión. El utilizar una base de datos permite estructurar la información para gestionar ésta de una forma adecuada y eficaz.

Los módulos de planificación y supervisión, pueden entenderse como sistemas basados en la planificación y supervisión propuestas en los Capítulos 3 y 4, respectivamente, por lo que no se entrará en más detalle, salvo en la sección 5.2 para mencionar los sistemas concretos empleados en el proyecto RHEA. Del mismo modo, el desarrollo del módulo de procesamiento de datos se escapa a los objetivos de esta tesis, por lo que tampoco se dará una explicación más detallada, salvo en la sección 5.2 para mencionar el sistema concreto empleado en RHEA. Los módulos restantes, es decir, los controladores de misión y el despachador, son por tanto los únicos que quedan por tratar para conseguir la automatización completa de la ejecución del sistema inspección/tratamiento. Estos módulos se describen a continuación, en las siguientes secciones.

5.1.1. Controladores de misión

Los controladores de misión que se proponen son los encargados de establecer la comunicación con las unidades y de proporcionar una interfaz para que los sistemas superiores puedan gobernar las flotas en una misión mediante órdenes de alto nivel tales como envío de planes, pausas, reanudaciones, paradas, etc.

En concreto, el controlador debe encargarse de descomponer el plan que contiene la misión conjunta en secuencias de órdenes más sencillas y entendibles por cada una de las unidades por separado. Por ejemplo, si se desea ejecutar una misión en una flota, la secuencia podría ser: 1) conectar con cada una de las unidades, 2) inicializarlas para la misión, por ejemplo, fijando el origen de coordenadas que debe usarse en las trayectorias contenidas en el plan y que tiene que ser el mismo para todas las unidades, 3) enviar los planes, 4) comprobar que se han recibido satisfactoriamente, 5) esperar que se lleven a cabo, 6) parar las unidades, 7) confirmar que efectivamente se han parado y 8) cerrar las conexiones.

Es necesario intercalar comprobaciones intermedias para confirmar que las órdenes se reciben y ejecutan, ya que en caso contrario podría existir un peligroso desfase, entre el estado teórico de la misión y el real. Por ejemplo, tras una orden de parada, se podría suponer que las unidades ya se han detenido y, sin embargo, no ser así a causa de algún error no detectado.

En definitiva, los controladores de misión propuestos automatizan la ejecución de las misiones construyendo una capa de alto nivel a partir de las operaciones básicas de control remoto que proporcionan las unidades. De este modo permiten salvar el hueco existente entre la planificación y la supervisión, y cerrar la automatización a alto nivel del ciclo de trabajo planificación-ejecución-supervisión.

Debido a que las secuencias de automatización de los controladores de misión dependen de los eventos originados por muchos sistemas, tales como peticiones desde la GUI normal o la portátil, *timeouts* (señales que indican que se ha sobrepasado el tiempo máximo de espera) en las comunicaciones, confirmaciones desde las unidades, errores de conexión, etc., y por eventos anteriores, tales como si ya se ha realizado una petición similar, si ya se está conectado, si se recibió respuesta, etc., se puede decir que estos controladores son sistemas *reactivos*. Los sistemas reactivos (Harel & Pnueli, 1985) son sistemas más difíciles de predecir que los *transformacionales* porque están guiados por multitud de eventos asíncronos y porque, además, no solo reaccionan a los estímulos externos a partir de los propios estímulos de ese instante sino también en función del estado adquirido a partir de los estímulos previamente recibidos, mientras que los sistemas *transformacionales* aplican una transformación a los datos de entrada de tal modo que suele bastar con una relación entre la entrada y la salida para especificarlos de una forma rigurosa y clara; los planificadores o el módulo de procesamiento de datos son ejemplos de estos últimos. Los comportamientos de los sistemas reactivos son complejos de analizar lo que hace que sean muy propensos a fallos. Por tanto, es esencial realizar un diseño cuidadoso especialmente cuando interactúan con elementos potencialmente peligrosos tal como hacen los controladores de misión que interaccionan con tractores que pueden llegar a pesar con los aperos cerca de la tonelada. Algunos ejemplos de este tipo de sistemas son los sistemas de control de procesos, los sistemas operativos de los ordenadores, los sistemas de comunicaciones, los sistemas de aviación, etc.

El principal problema en el diseño de los sistemas reactivos radica en la dificultad que entraña especificar formal y rigurosamente las conductas reactivas de una forma

clara, a diferencia de los sistemas transformacionales que, incluso en los casos más complejos, se pueden descomponer en partes más pequeñas fácilmente especificables. Aunque es posible expresar los sistemas reactivos mediante secuencias de eventos de entrada/salida y condiciones/acciones, no es fácil concretarlos de una forma fácilmente entendible. Las especificaciones habituales basadas en separaciones en niveles y módulos no son suficientes por el inmenso número de situaciones posibles, especialmente cuando los comportamientos están compuestos por muchos pasos, interaccionan con muchos sistemas y dependen de eventos anteriores. Por ello, es habitual emplear máquinas de estados para diseñar y desarrollar este tipo de sistemas (Harel, 1987), ya que la riqueza semántica de éstas permite representaciones visuales fácilmente entendibles e incluso existen herramientas capaces de generar código automáticamente a partir de las propias especificaciones suministradas como máquinas de estados (The State Machine Compiler, 2015). Por todo esto, se decidió usar máquinas de estados basadas en el modelo *statechart* (Harel, 1987) para especificar los controladores que manejan la flota.

El modelo *statechart* utiliza la notación tradicional de representar los estados mediante rectángulos redondeados y las transiciones con flechas, pero es más rico semánticamente y permite también súper-estados (estados con otros estados dentro) y transiciones hacia estados pasados dentro de los súper-estados. Los súper-estados son útiles para agrupar transiciones similares hacia un mismo estado destino. Las transiciones hacia estados pasados son útiles cuando se desea volver al último estado dentro de un súper-estado pero no se sabe cuál era exactamente.

La Figura 5.2 muestra la utilidad de estas dos nuevas herramientas comparando una misma máquina de estados con la notación tradicional con otra con el modelo *statechart*. Los estados *Movimiento*, *Pausado* y *Parado* representan varios estados de una unidad, desde los que es posible activar el apero. Con la notación tradicional tienen que usarse tres estados *Activando Aperó* diferentes para poder regresar al estado de origen. En el modelo *statechart* es suficiente el estado “H” (por “*History*” en la notación *statechart*), que representa el último de los estados en el que se estuvo dentro de ese súper-estado. Esto hace que solo sea necesario un único estado destino *Activando Aperó*, porque ya no hay tres posibles estados de vuelta. Como se observa la máquina de estados es más clara y se han reducido considerablemente el número de transiciones.

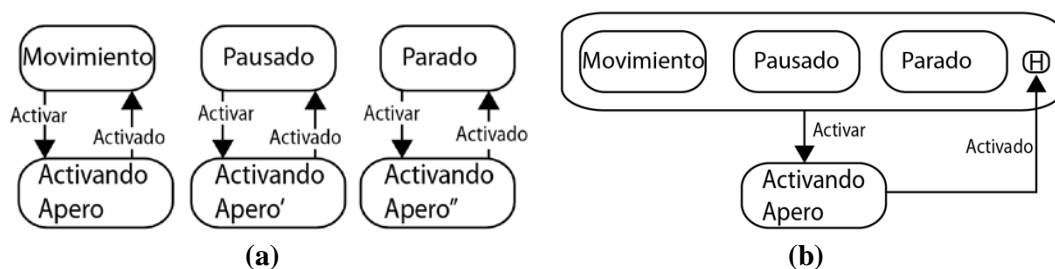


Figura 5.2. Diagrama de estados necesario para poder activar el apero en una unidad desde cualquiera de los estados principales (a) con una representación tradicional y (b) usando la representación *statechart*

En definitiva, si se supone que las unidades son capaces de aceptar conexiones externas y que están lo suficientemente automatizadas como para disponer de un pequeño conjunto de operaciones básicas de control remoto, tales como las que se muestran en la

Tabla 5.1, el diagrama de estados para un controlador de misión de flota genérico basándose en la representación *statechart* sería el que se muestra en la Figura 5.3

Tabla 5.1. Operaciones básicas proporcionadas por las unidades y que se han considerado para diseñar los controladores de misión.

Operación	Descripción
Inicialización	Carga de la configuración inicial de la unidad
Actuaciones	Acciones sobre la propia unidad (desplazamiento, velocidad, activación del apereo, seguimiento de un plan...)
Pausa	Interrupción de la operación que se esté ejecutando en ese momento y permanencia en ese estado sin posibilidad de salir hasta que se envíe una orden de reanudación.
Reanudación	Reanudación de la actividad en la unidad y de la operación que se había dejado en pausa.
Parada	Parada total de movimiento de la unidad y de cualquiera de sus dispositivos
Cerrar conexión	Cierra la conexión desde la que se ha realizado la petición

Las operaciones consideradas (véase Tabla 5.1) son relativamente habituales en drones y robots de tierra, y son una generalización de las consideradas por el Sistema de Control de Unidad expuestas en el Capítulo 4 (véase Tabla 4.3).

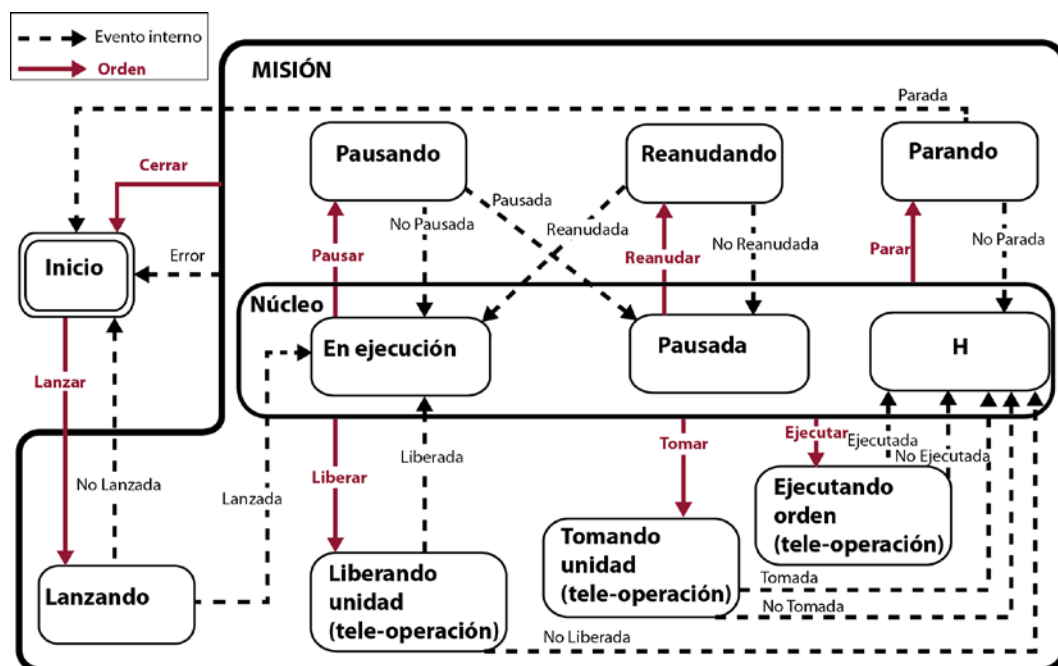


Figura 5.3. Diagrama de estados de un controlador de misiones con flotas de vehículos

Las órdenes de alto nivel que acepta este controlador y que figuran en rojo en la Figura 5.3 son las siguientes:

- Lanzar misión: El controlador establece la comunicación con las unidades y envía el plan correspondiente a cada una. En caso de que no se consiga que ninguna de las unidades reciba y comience a ejecutar el plan, el propio controlador emite la señal de *No Lanzada* y se vuelve al estado inicial. Por el contrario, basta con que una sola unidad comience a ejecutar del plan asignado para que el controlador emita la señal lanzada y transite al estado *En Ejecución*.
- Pausar misión: El controlador ordena a todas las unidades que se pausen, ya que solo si todas las unidades se paran se puede considerar que la misión está pausada. En caso de que todas cumplan con la orden el controlador pasa al estado *Pausada*. Si alguna unidad sigue en movimiento el controlador emite la señal *No Pausada* y vuelve al estado *En ejecución*.
- Reanudar misión: El controlador ordena a todas las unidades que retomen la misión que habían dejado en pausa. Basta con que una sola unidad reanude la marcha para que el controlador vuelva al estado *En ejecución*. Si por el contrario ninguna unidad reanuda la marcha, el controlador vuelve al estado *Pausada*.
- Parar misión: El controlador envía a todas las unidades la orden de parar por completo, sin posibilidad de recuperar la misión. Si todas las unidades cumplen la orden, la misión finaliza, se cierran las conexiones y el controlador vuelve al estado inicial. En caso contrario el controlador vuelve al estado en el que estaba.
- Cerrar conexiones: Esta orden se encarga de cerrar todas las conexiones abiertas cuando las unidades ya están paradas. Cuando el Supervisor detecta el fin de la misión, lo notifica a los módulos superiores (despachador y GUI) y son estos quienes deciden cerrar las comunicaciones con la flota.
- Tomar unidad (para tele-operación): Esta orden permite tomar el control sobre una unidad de la flota que está participando en la misión. Para ello, el controlador almacena la misión que le queda por cumplir a la unidad, extrayéndola del *supervisor de misión restante* y, a continuación, para la unidad. Esta orden permite que un operario pueda tele-operar una unidad durante la misión, por ejemplo para comprobar su funcionamiento una vez detectado un problema de funcionamiento. Una vez que un operario toma el control de una unidad, el controlador la deja de considerar como parte de la flota de cara a las órdenes de pausa y reanudación, así como para determinar si la misión está pausada o en ejecución.
- Liberar unidad (de tele-operación): Esta orden devuelve la unidad a la misión. Para ello el controlador para la unidad, toma el control de la misma y restablece la misión que le quedaba a la unidad por realizar y que el controlador había almacenado previamente.
- Ejecutar orden(de tele-operación): Esta orden le permite a un operario en la estación base tele-operar las unidades que se han apartado de la misión enviando directamente órdenes de desplazamiento, ajuste del apero, etc.

Asimismo, puesto que muchas de las órdenes actúan sobre toda la flota (lanzar, pausar, reanudar, parar y cerrar), el controlador debe llevar un registro de la evolución de la unidades, una vez que ha enviado los comandos, para determinar cuándo se producen las señales o eventos internos que cambian el estado de la misión, líneas discontinuas en la Figura 5.3. Por ejemplo, para saber si se ha lanzado la misión es necesario saber si al

menos una de las unidades ha conseguido empezar adecuadamente, o para que la misión esté en pausa es necesario saber si todas las unidades están en pausa, por el contrario, para que la misión se reanude basta con que una de sus unidades vuelva a modo ejecución. Asimismo, solo si todas las unidades están detenidas se puede considerar que la misión está parada.

Algunas de las órdenes del controlador propuesto, internamente no pueden traducirse en un único comando básico (véase Tabla 5.1) para enviar a las unidades, sino que deben descomponerse en una secuencia de varios comandos. Éste es el caso de las operaciones *Lanzar misión* y *Liberar unidad*, que se traducen en dos peticiones; una primera de inicialización de la unidad y una segunda de envío del plan de la misión. En la operación *Lanzar* es necesario el paso de inicialización para fijar por primera vez la configuración inicial de la unidad, es decir, el origen de coordenadas, la hora, la posición de la estación base, etc. Del mismo modo, en la operación *Liberar* también se necesita este primer paso porque se viene de una fase de tele-operación en la que se puede haber modificado deliberadamente estos parámetros, por lo que hay que asegurar que se restauran los valores primitivos, y un segundo paso en el que se envía la parte del plan que le quedaba a la unidad antes de que fuera liberada para ser controlada remotamente. Por ello, es necesario crear una capa de más bajo nivel formada por tres controladores, uno por unidad, llamados controladores de unidades en misión sobre los que se apoya el controlador de misión. Estos controladores están encargados de interpretar las órdenes que el controlador de misión solicita a cada una de las unidades durante una misión y codificarlas a partir de las órdenes básicas que admiten las unidades (véase la Tabla 5.1). La Figura 5.4 muestra el diagrama de estados de uno de estos controladores.

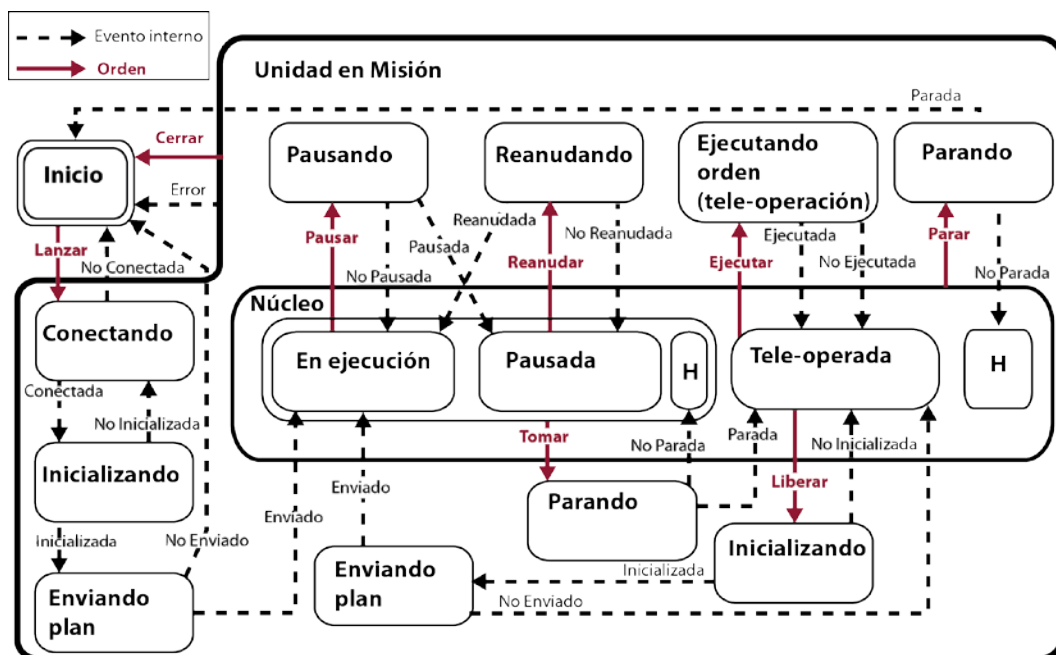


Figura 5.4. Diagrama de estados del controlador de una unidad en misión

En la Figura 5.4 se muestran en rojo las operaciones que estos controladores proporcionan, que son las mismas que las del controlador de misión, solo que éstas actúan sobre una única unidad, en lugar de sobre toda la flota. La máquina de estados es muy parecida a la de la Figura 5.3. La principal diferencia es que las señales internas, en línea discontinua, ya no dependen de toda la flota, sino solo de la unidad manejada. Además, las operaciones de *Lanzar misión* y *Liberar unidad* se han separado en varios pasos para poder expresarlas en términos de las operaciones de bajo nivel que pueden ejecutar las unidades. Ahora cada estado se identifica claramente con cada una de las operaciones consideradas.

A todo lo anterior hay que añadir la necesidad de crear una última capa que proporcione un servicio de confirmación de la ejecución de los órdenes, ya que las unidades generalmente solo son capaces de confirmar la recepción de los comandos sin garantizar su ejecución. Para ello el controlador de unidad en misión debe apoyarse en el controlador básico de unidad que se muestra en la Figura 5.5. Este controlador construye el servicio de confirmación basándose en dos hechos: 1) que las unidades son capaces de notificar la recepción de las órdenes mediante un ACK (del inglés *acknowledgement*, en español acuse de recibo), mensajes muy comunes en multitud de protocolos de comunicación (ACK, 2015), y 2) que las unidades emiten periódicamente cierta información de estado, información que se utiliza en la supervisión. Para cada posible petición, se comprueba primero si se recibe el ACK y, en caso afirmativo, se espera a que los mensajes de información de estado desde la unidad corroboren que la orden se ha ejecutado. En caso de que pase un determinado tiempo sin recibirse el ACK o sin detectarse en la información de estado el cambio esperado, salta un *timeout* y el controlador vuelve al estado en el que estaba. Las transiciones que aparecen en rojo en la Figura 5.5 son las operaciones que proporciona este controlador y que se corresponden con los comandos básicos mostrados en la Tabla 5.1.

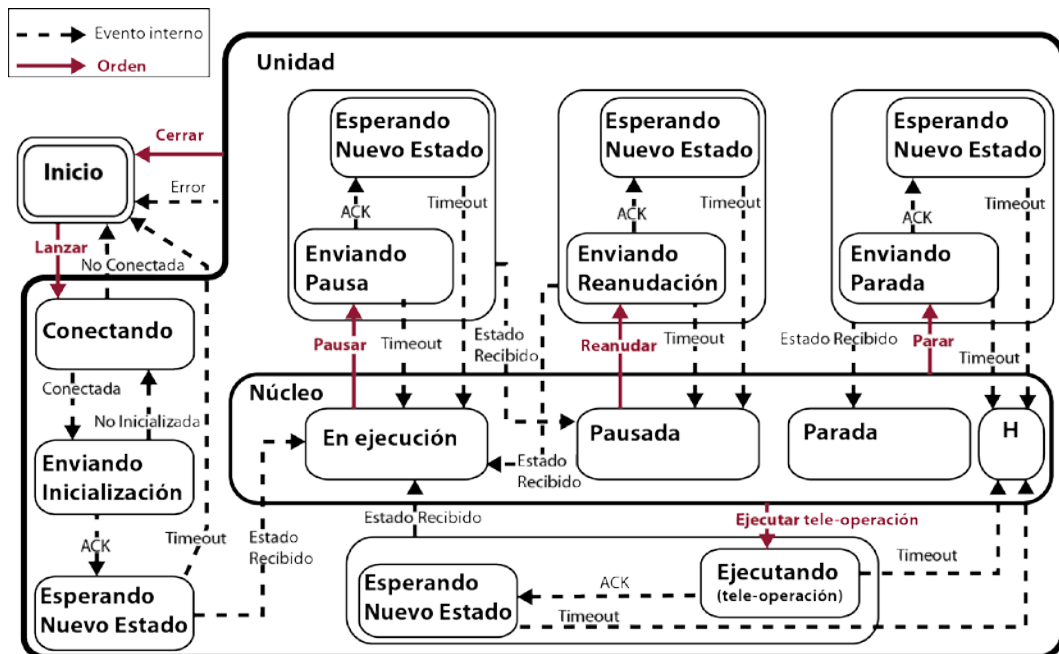


Figura 5.5. Diagrama de estados del controlador básico de unidad

Mediante esta separación en capas de diferente complejidad es posible construir un controlador de misiones genérico de alto nivel, capaz de gobernar una flota de varias unidades con facilidad y precisión de respuesta, este último aspecto gracias al servicio de confirmación incluido en la propuesta.

Además gracias a la clara separación en estados que permiten los diagramas *statechart*, es inmediato saber si se debe descartar una petición o no a partir de un estado dado; por ejemplo, no debería procesarse nunca una petición de pausa salvo que se esté en el estado *En ejecución*.

5.1.2. Despachador

El despachador es el módulo distribuidor del Gestor de Misiones. Contiene la lógica necesaria para redirigir las peticiones que llegan desde la GUI o el dispositivo portátil y para crear el canal de comunicación con el sistema destinatario. Concentra todas las comunicaciones con los sistemas internos del Gestor, lo que es útil porque se crea una capa de abstracción que permite sustituir cualquiera de estos sistemas sin que ninguna de las interfaces tenga que realizar ningún cambio.

Al igual que los controladores, el funcionamiento del despachador puede expresarse mediante una máquina de estados para especificar mejor qué peticiones pueden ejecutarse en cada momento y cuáles deben descartarse. La Figura 5.6 muestra el diagrama de estados propuesto para el despachador.

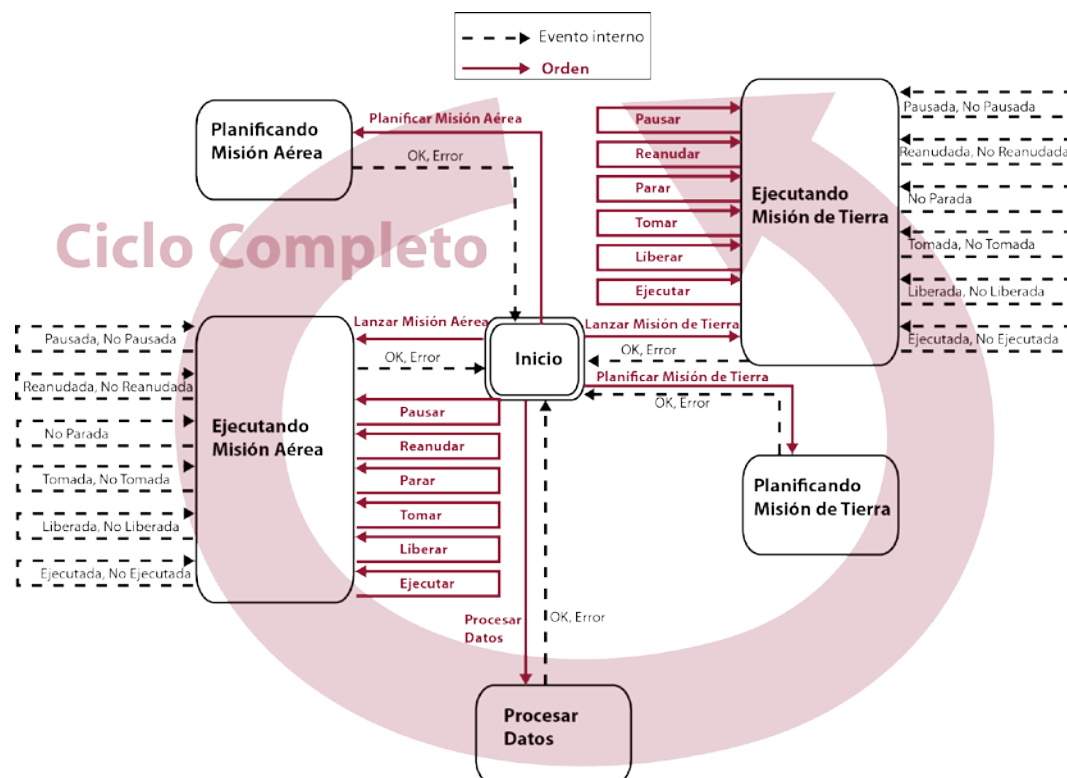


Figura 5.6. Diagrama de estados del despachador que incluye todas las peticiones que se han considerado

Dentro de los estados asociados a los planificadores (aéreo y terrestre) y al procesamiento de datos, basta con llamar a los sistemas correspondientes y esperar la respuesta de estos. Por el contrario, dentro de los estados asociados a misiones, se emplean tanto los controladores como los supervisores de misión. Toda la información suministrada por los controladores se redirige a los supervisores para que la analicen y detecten posibles anomalías.

Obsérvese que si se ejecuta en orden el ciclo completo es posible combinar misiones aéreas y terrestres y acometer, con ello, los escenarios descritos en el Capítulo 1.

5.2. Implementación para RHEA

El Gestor de Misiones implementado para RHEA se basó en el modelo genérico planteado en el apartado anterior. Para ello integró un planificador aéreo (Valente, Del Cerro, Barrientos, & Sanz, 2013) y un controlador de misión y supervisor aéreo (Del Cerro, Sanz, Valente, Rossi, Cancar, & Barrientos, 2014) desarrollados por la Escuela Técnica Superior de Ingenieros Industriales de la UPM. Estos módulos eran capaces de manejar la flota aérea de RHEA compuesta por dos drones de seis rotores (modelo AR200, véase Figura 5.7a) desarrollados por la compañía AirRobot (AirRobot Company, 2015), capaces de transportar una carga de hasta 1,5 kg con una autonomía de vuelo de aproximadamente 40 minutos. Los drones se equiparon con dos cámaras, de espectro visible e infrarrojo cercano, montadas en un sistema de suspensión cardán para poder orientarlas hacia abajo (véase Figura 5.7b).

Los drones eran unidades lo suficientemente automatizadas como para aceptar planes formados principalmente por una lista de puntos (*waypoints*) por los que los drones debían pasar y tomar fotos de forma totalmente autónoma. Además los vehículos aéreos eran capaces de proporcionar información de telemetría durante el vuelo, necesaria para la supervisión, así como estimaciones de la posición en la que estaban y del nivel de la batería.



Figura 5.7. (a) Drone AR200 en vuelo y (b) cámaras con las que estaban equipadas los drones

Para las misiones terrestres de tratamiento, se integraron tres módulos, a saber, un planificador, un supervisor y un controlador de flota terrestre. Los dos primeros estaban basados en los sistemas de planificación y supervisión propuestos en este trabajo de investigación y descritos en los Capítulos 3 y 4. El tercer módulo se implementó a partir

del controlador de flota y los diagramas propuestos en el apartado anterior (5.1.1). Para ello se utilizó el entorno de máquinas de estados proporcionado por las librerías Qt. Los tres módulos eran genéricos y se ajustaron para trabajar con la flota de tierra de RHEA, descrita en la sección 4.2.

Como, además, el sistema global se probó resolviendo uno de los escenarios descritos en la introducción, en concreto la eliminación de malas hierbas en trigo mediante pulverización, se integraron como módulo de procesamiento de datos, los sistemas de mosaicado (Rabatel & Labbé, 2015) y cartografía (detección y geo-referenciación) de malas hierbas (Peña-Barragán, Torres-Sánchez, de Castro, Kelly, & López-Granados, 2013) desarrollados por IRSTEA (Francia) y el Instituto de Agricultura Sostenible del CSIC, respectivamente. El objetivo del sistema de mosaicado era componer y ortorectificar todas las fotos tomadas durante la misión aérea, produciendo en una única imagen de todo el campo, y el sistema de cartografía de malas hierbas estaba encargado de procesar la imagen generada por el sistema de mosaicado y encontrar la posición de las malas hierbas a partir del contorno del campo, el tipo de cultivo y la dirección de las líneas.

Por último, el módulo despachador también se implementó a partir del diagrama de estados propuesto en el apartado 5.1.2, utilizando el entorno para máquinas de estados de Qt.

5.2.1. Interfaz GUI y Comunicaciones entre módulos

Los sistemas que requerían una comunicación puntual y que no se ejecutaban en un equipo distinto al del Gestor se integraron mediante librería, este es el caso de los planificadores. Por el contrario, los sistemas como los controladores de misión y los supervisores que requerían una comunicación continua, para informar periódicamente de su estado, se integraron mediante *sockets*. También se integraron de este modo los sistemas de mosaicado y cartografía ya que se ejecutaban en máquinas diferentes por motivos internos del proyecto que tenían que ver con las licencias de algunos de los programas utilizados por los grupos del IRSTEA y del Instituto de Agricultura Sostenible del CSIC.

Para las comunicaciones entre las interfaces y el Gestor también se emplearon *sockets*, para permitir que éstas pudieran estar instaladas en otros equipos, como en el caso de la interfaz portátil. La comunicación a través de los *sockets* se codificó en mensajes con el formato mostrado en la Tabla 5.2.

Tabla 5.2. Formato de mensaje

Delimitador	Tamaño	TimeStamp	Código	Carga
(1 byte)	(2 bytes)	(8 bytes)	(1 byte)	(0– 65535 bytes)

El *delimitador* es un byte acordado (0x7E, en el caso de RHEA) que siempre marca el inicio del mensaje. Este valor no puede aparecer en el resto del mensaje, por lo que es necesario aplicar un procedimiento de escape de carácter siempre antes de enviar el mensaje. En RHEA se sustituyó el byte 0x7E por dos bytes: 0x7D y 0x5E, aunque antes se había reemplazado el byte 0x7D por los bytes 0x7D y 0x5D, para que de este modo el mensaje no pudiera contener nunca secuencias 0x7D5E. Cuando se recibía el mensaje,

antes de extraer los campos, había que reemplazar primero la secuencias 0x7D5E por el byte 0x7E y después la secuencia 0x7D5D por 0x7D. La Tabla 5.3 muestra un ejemplo de un mensaje modificado siguiendo el procedimiento planteado.

Tabla 5.3. Ejemplo de mensaje con carácter de escape sustituido.

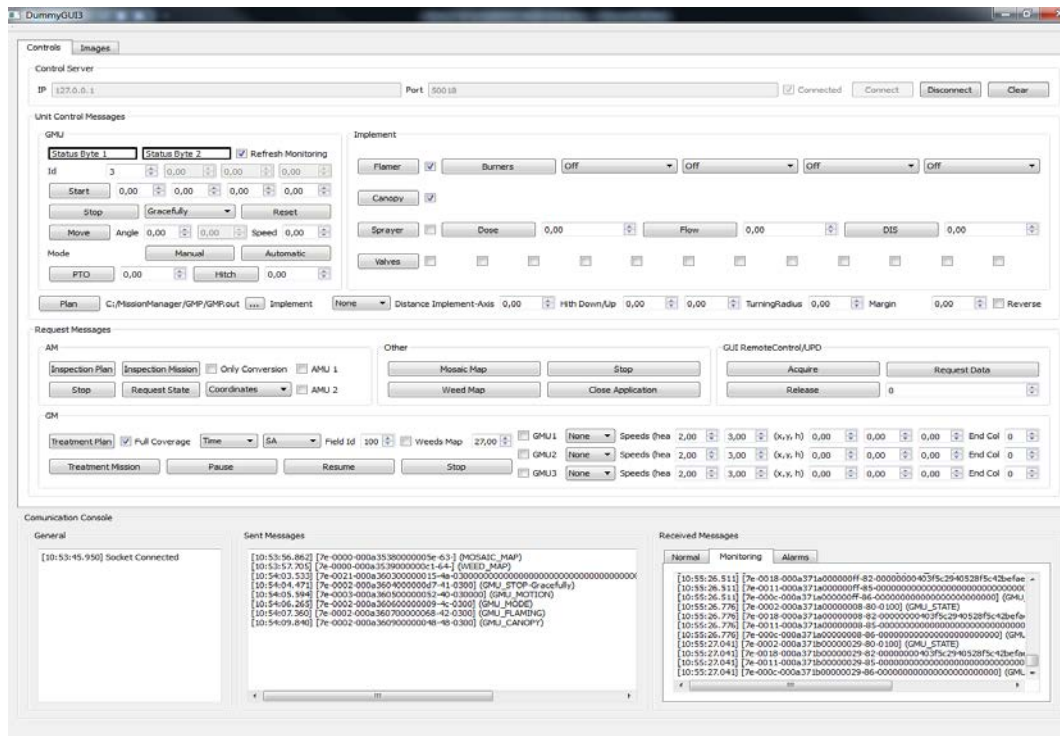
Mensaje original	0x7E 0x01 0x07D 0x5E 0x02 0x03 0x04 0x05 0x06 0x01 0x7E
Mensaje modificado	0x7E 0x01 0x07D 0x5D 0x5E 0x02 0x03 0x04 0x05 0x06 0x01 0x7D 0x5E

El campo *tamaño* contiene el número de bytes que ocupa el campo *carga*. El *timestamp* contiene la fecha de emisión del mensaje. El código indica el tipo de mensaje, es decir, si codifica un *Lanzar misión*, una solicitud de mosaicado, una pausa, un mensaje de monitorización que proviene de las unidades, una alarma generada en alguno de los supervisores, etc. Por último, la *carga* lleva la información necesaria para ejecutar la petición codificada en el mensaje. Por ejemplo, si es un mensaje de envío de plan, la carga contiene la trayectoria, las velocidades, el estado de los aperos, etc., en definitiva el plan para un robot codificado.

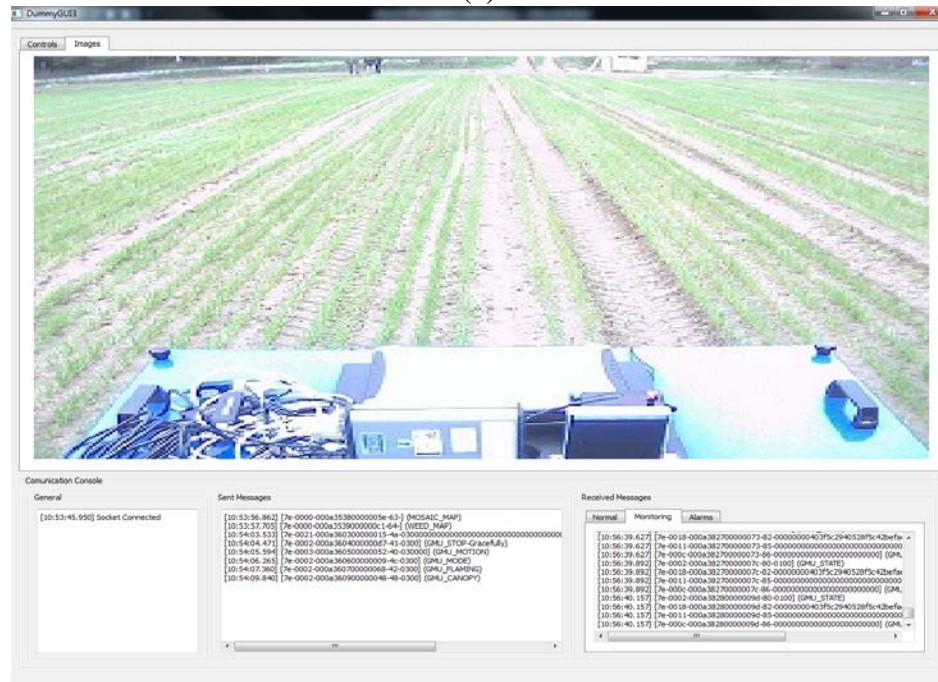
Para poder acceder a los diferentes servicios del Gestor de Misiones se desarrolló una interfaz GUI usando las librerías Qt. La interfaz se servía de un socket para enviar peticiones basadas en el formato de mensaje mostrado en la Tabla 5.2. Las peticiones más importantes que se podían realizar eran las mostradas en la Figura 5.6, es decir, planificar una misión (aérea/terrestre), ejecutar una misión (aérea/terrestre), procesar los datos adquiridos durante las misiones o lo que es lo mismos construir el mosaico y generar el mapa de distribución de malas hierbas, y controlar remotamente o liberar una unidad durante una misión.

Una vez enviadas las peticiones, la interfaz mantiene el canal abierto para recibir y mostrar al operario la información suministrada por el subsistema interno correspondiente. Por ejemplo, en el caso de ejecutar una misión, la interfaz permite monitorizar la misión en tiempo real visualizando por dónde se están moviendo las unidades gracias a las imágenes proporcionadas e integradas dentro de la información de monitorización (véase Tabla 4.1). Asimismo, también permite seguir los estados de las unidades con precisión y recibir las alarmas generadas. La Figura 5.8 muestra el aspecto de algunas de las ventanas de la interfaz.

En el caso de la petición de planificación, la interfaz invoca al Planificador construido según lo explicado en el Capítulo 3 y con los parámetros por defecto que allí se proponen. Aunque, para poder invocar al Planificador con cualquier tipo de método de optimización y configuración, se desarrolló otra interfaz con acceso total a cualquiera de los parámetros del Planificador, tal como se muestra en la Figura 5.9a esta misma interfaz también se puede seguir con mayor precisión la convergencia hacia la solución óptima con el método de optimización seleccionado (véase la Figura 5.9b) y el resultado final (véase la Figura 5.10).

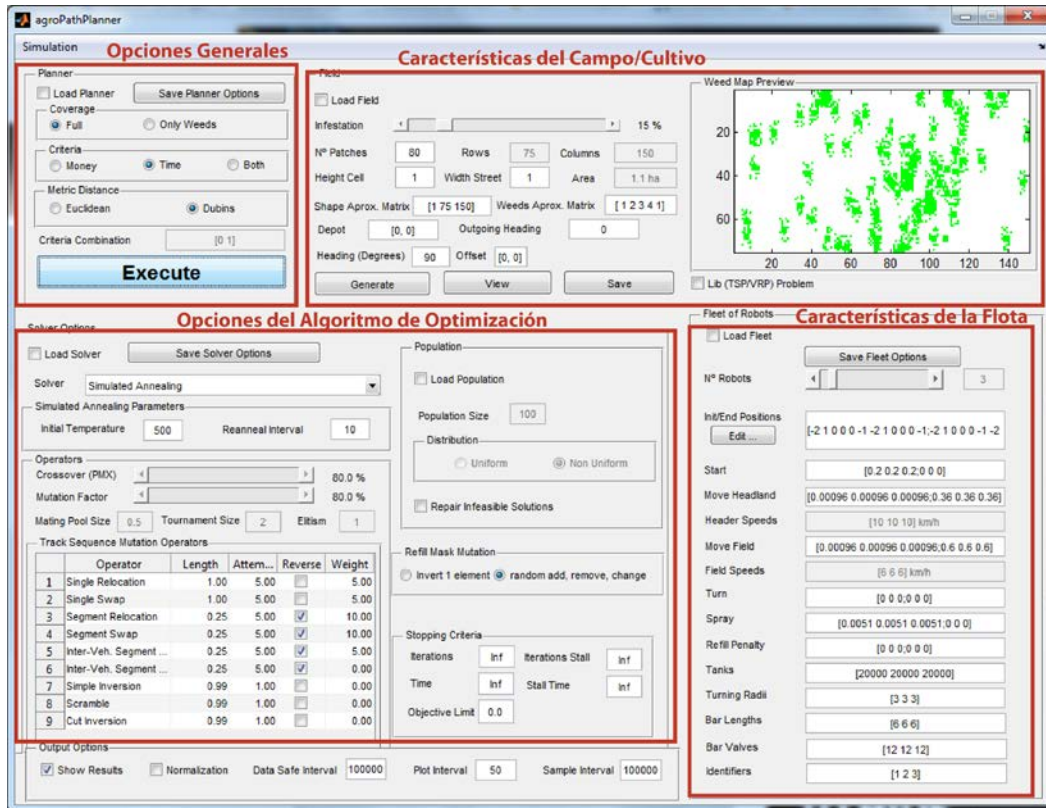


(a)

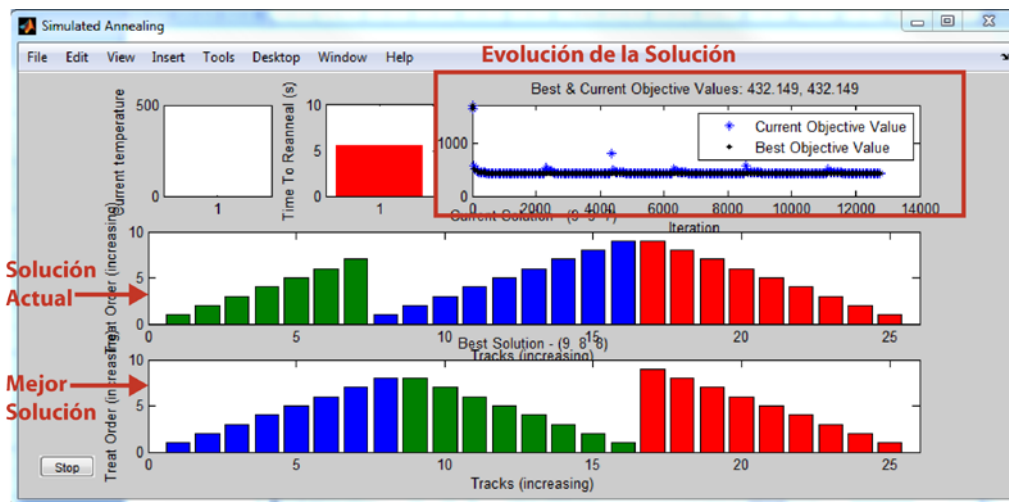


(b)

Figura 5.8. Interfaz GUI. (a) Panel para realizar peticiones (planificaciones, mosaicos, lanzar misiones, etc.), visualizar la información de monitorización de cada unidad (posición, velocidad, estado del apero) y tele-operar una unidad durante una misión. (b) Panel para visualizar las imágenes proporcionadas por la unidad durante la misión



(a)



(b)

Figura 5.9. Interfaz del Planificador: (a) Panel con acceso a todos los parámetros a configurar y (b) panel de evolución de la convergencia

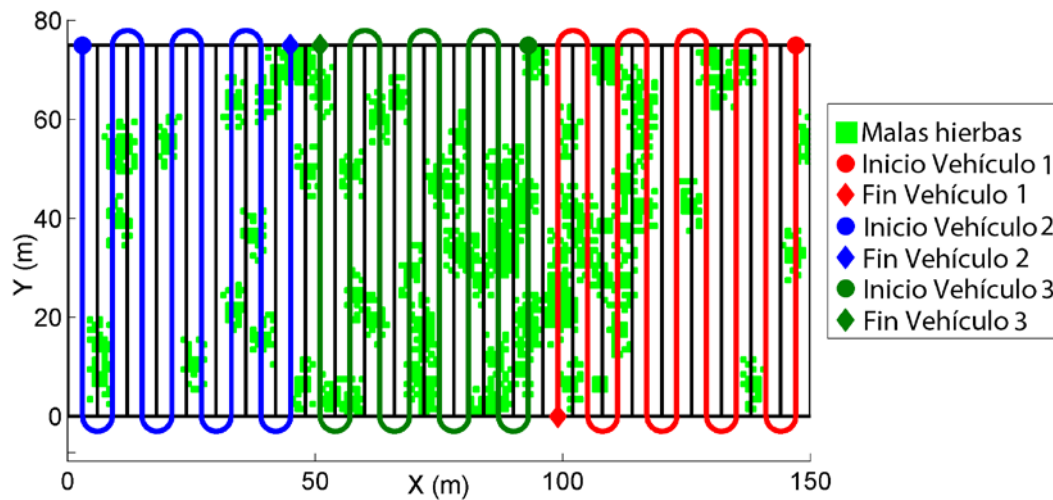


Figura 5.10. Interfaz del Planificador. Panel que muestra el resultado final de la planificación

5.3. Resultados

El sistema completo se evaluó durante una de las demostraciones del proyecto RHEA. El objetivo de la demostración era validar el escenario más completo de RHEA, es decir, aquél descrito en la introducción para el cual era necesaria la intervención de todos los sistemas incluidos en el Gestor de Misiones. Se trataba de la pulverización localizada con herbicida siguiendo un mapa de distribución de rodales de mala hierba en un cultivo de trigo obtenido de una misión previa de inspección. En concreto los pasos a realizar fueron los siguientes:

- 1) Inspección aérea del campo mediante drones para tomar imágenes cenitales del cultivo. Este paso se descompuso a su vez en dos tareas:
 - a. Planificación de las rutas a seguir por los drones para no dejar ninguna parte del cultivo sin fotografiar
 - b. Ejecución de la misión aérea siguiendo el plan generado en el paso anterior y con una supervisión capaz de generar alarmas siempre que se detectase algún tipo de problemas como, por ejemplo, nivel de batería bajo, riesgo de colisión, etc.
- 2) Generación del mapa de distribución de las malas hierbas presentes en el cultivo:
 - a. Mosaicado y ortorectificación de las imágenes tomadas por los drones generando una única fotografía cenital de todo el cultivo
 - b. Procesado de la imagen y detección de las zonas con malas hierbas. Generación del mapa geo-referenciando las zonas infestadas
- 3) Tratamiento del campo mediante tractores robotizados equipados con una barra de pulverización localizada, con capacidad para activar/desactivar independientemente cada una de sus boquillas:

- a. Planificación de las rutas a seguir por los tractores para no dejar ninguna parte del cultivo sin tratar siguiendo las estrategias propuestas en este trabajo de investigación y presentadas en el Capítulo 3.
- b. Ejecución de la misión de tratamiento, con una total supervisión de las unidades tal como se explicó en el Capítulo 4 de la presente memoria.

Gracias a lo completo del escenario fue posible probar de forma secuencial y automatizada todos los módulos descritos en el Gestor de Misiones propuesto en este capítulo. Todo el escenario pudo ser ejecutado automáticamente a través de una interfaz GUI instalada en la estación base, similar a la explicada en el apartado anterior, que remitía las peticiones oportunas al Gestor de Misiones a través del despachador. Los únicos pasos que requirieron la intervención directa del operario fueron: 1) la transferencia de las fotos desde las tarjetas de las cámaras a bordo de los drones a los ordenadores de la estación base para descargar las fotos y 2) el despegue y aterrizaje de los drones, que tuvo que ser realizado por un piloto, debido a la legislación española vigente.

Para la realización de esta prueba, se preparó en la finca experimental del CSIC “La Poveda” en Arganda del Rey (40°18'51.102"N, 3°29'03.379"W), un campo de pruebas de 2.400 m² con trigo de invierno siguiendo las prácticas comunes de la zona. Simultáneamente, se sembraron nueve rodales de malas hierbas de 3 m x 3 m con semillas de mostaza (*Sinapis arvensis*) manteniendo el resto del campo libre sin infestación mediante la aplicación de un tratamiento de herbicida de pre-emergencia y otro de post-emergencia que respectaba los rodales de malas hierbas sembrados mediante la colocación de unas lonas de plástico que los cubrían, tal como se muestra en la Figura 5.11.



Figura 5.11. Cultivo de trigo de invierno donde se han protegido con unas lonas de plástico 9 rodales de malas hierbas que se habían sembrado para realizar las pruebas.

En primer lugar se tomaron manualmente las posiciones GPS de los cuatro puntos que definían el contorno del campo (rectángulo amarillo en la Figura 5.12) y se incluyó esta información en la base de datos del sistema. Posteriormente, a través de la GUI, se solicitó una planificación para una misión aérea de inspección sobre el campo delimitado por ese contorno. El planificador definió una región de seguridad, dilatando el contorno del campo, que no podía ser sobrevolada por las unidades aéreas (contorno verde en la

Figura 5.12). Una vez fijada esta zona de seguridad, a partir de la resolución de las cámaras, la altura y el porcentaje de solapamiento deseado en las imágenes, el planificador generó las rutas con los puntos intermedios (*waypoints*) en los que debían tomarse las fotografías (trayectorias roja y azul en la Figura 5.12).

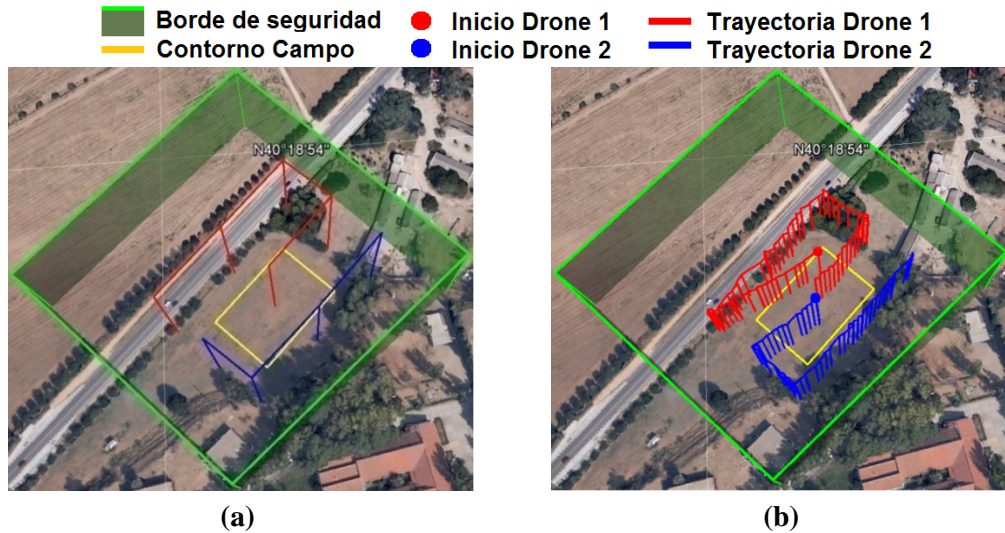


Figura 5.12. Misión aérea de inspección: (a) planificación y (b) ejecución con rutas reales

Una vez insertadas manualmente las imágenes en el sistema, se solicitó su mosaicado a través de la GUI que resultó en la imagen mostrada en la Figura 5.13a. La imagen fue procesada para discriminar las zonas con malas hierbas (véase Figura 5.13b) y construir el mapa de distribución de rodales de malas hierbas que se utilizó posteriormente en la planificación de la misión de tratamiento que llevaron a cabo las unidades terrestres.

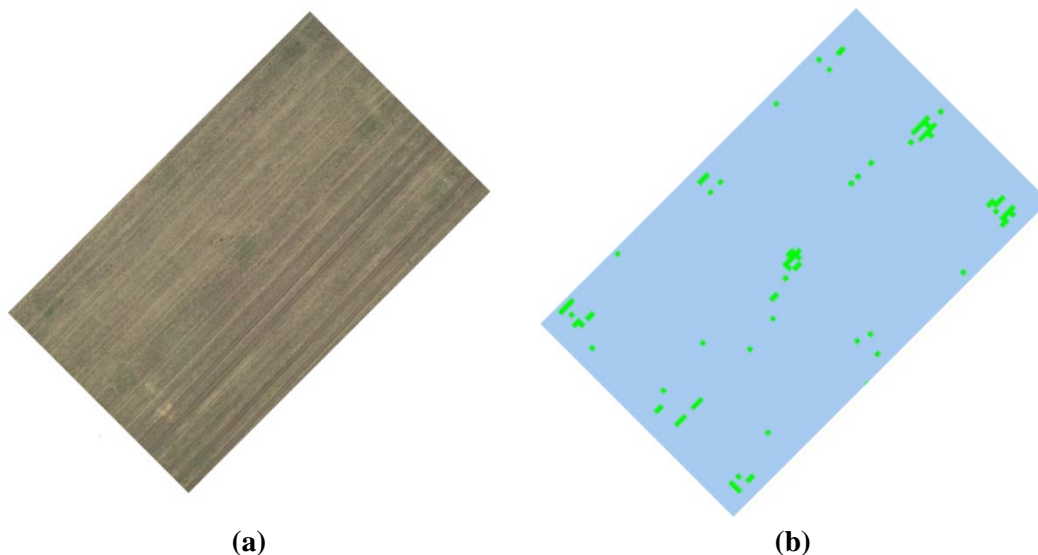


Figura 5.13. Generación del mapa de malas hierbas: (a) imagen mosaicada y (b) malas hierbas mapeadas sobre el campo

Para la misión de tratamiento, los rodales y otras partes del campo se cubrieron con tiras de papel y el herbicida se tiñó con colorante para detectar visualmente si se pulverizaba con precisión en las zonas esperadas (véase Figura 5.14a).

No obstante, como las malas hierbas no habían crecido hasta formar el rodal previsto (véase Figura 5.14b), el sistema de cartografía no generó el mapa con los rodales rectangulares sembrados tal como se muestra en la Figura 5.13b. Como sobre los rodales reales era difícil comprobar si la pulverización era precisa o no, se geo-referenciaron los rodales sembrados a partir de las 4 estacas que los definían y que se habían colocado en el momento de la siembra y se construyó, mediante un proceso semi-automático, el mapa que aparece en la Figura 5.15a y que fue el que se utilizó para planificar la misión terrestre.



Figura 5.14. (a) Tiras de papel dispuestas a lo largo del campo para verificar la pulverización y (b) rodal real versus rodal sembrado.

Como se observa en la Figura 5.15b, la superficie que se pulverizó encaja con bastante precisión con los rodales especificados. La parte derecha del campo se reservó para hacer otras pruebas, por eso la trayectoria mostrada en la Figura 5.15 solo cubre 6 rodales o lo que es lo mismo la parte izquierda del cultivo.

En total se pulverizó más del 97% de las áreas objetivo (rodales de malas hierbas). Las diferencias entre las áreas pulverizadas y las áreas objetivo se debieron a ligeros retrasos en la apertura y cierre de las boquillas en los bordes de los rodales (Figura 5.16a). El retardo de apertura produjo un desfase medio de 22,8 cm con una desviación típica de 27,8 mientras que el de cierre de -2,4 cm, con una desviación típica de 48,7 cm. Estos desfases fueron detectados y notificados en forma de alarma al operario a cargo de la misión.

En cuanto a las trayectorias, para medir la desviación del plan de ruta establecido por el Planificador, la trayectoria objetivo o de referencia se marcó con una cinta blanca de 80 m que conectaba dos extremos de la parcela. La trayectoria realmente seguida por el vehículo se extrajo de la huella producida por las ruedas durante el recorrido. Se realizaron dos mediciones en dos momentos diferentes, registrando la distancia entre trayectorias (referencia y real) en 24 puntos separados 3 m (prueba 1 en Figura 5.16a) y en 16 puntos separados 5 m (prueba 2). La Figura 5.16b muestra que el vehículo autónomo siguió con gran precisión la trayectoria sin apenas desviarse del plan de ruta establecido. Se observaron desviaciones inferiores a 7 cm en las medidas comprendidas

entre los percentiles 25 y 75 respecto a la trayectoria definida en las dos pruebas realizadas.

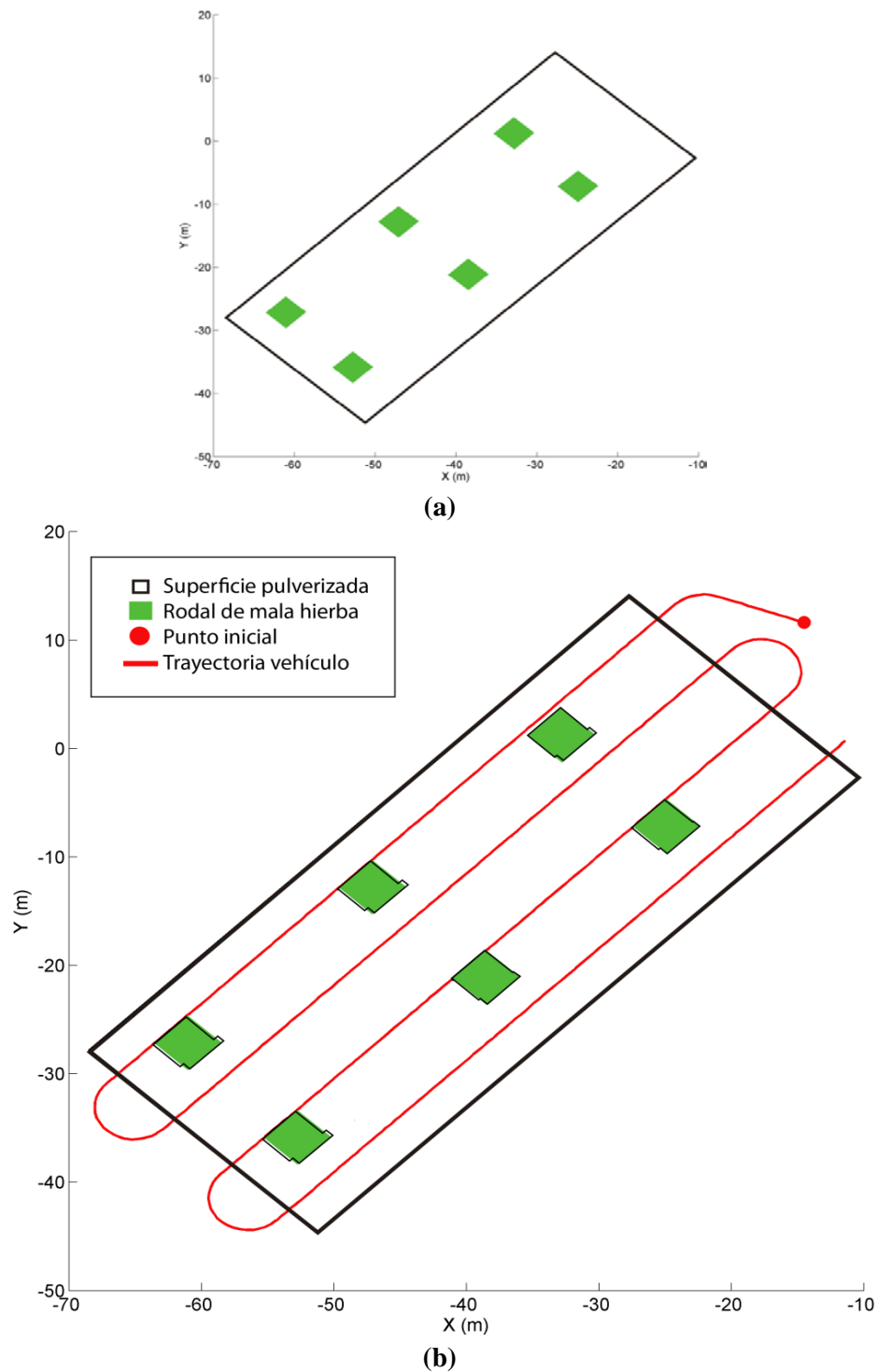


Figura 5.15. (a) Mapa de malas hierbas empleado para la misión de tratamiento realizada por las unidades terrestre y (b) trayectoria de la unidad con la superficie que pulverizó

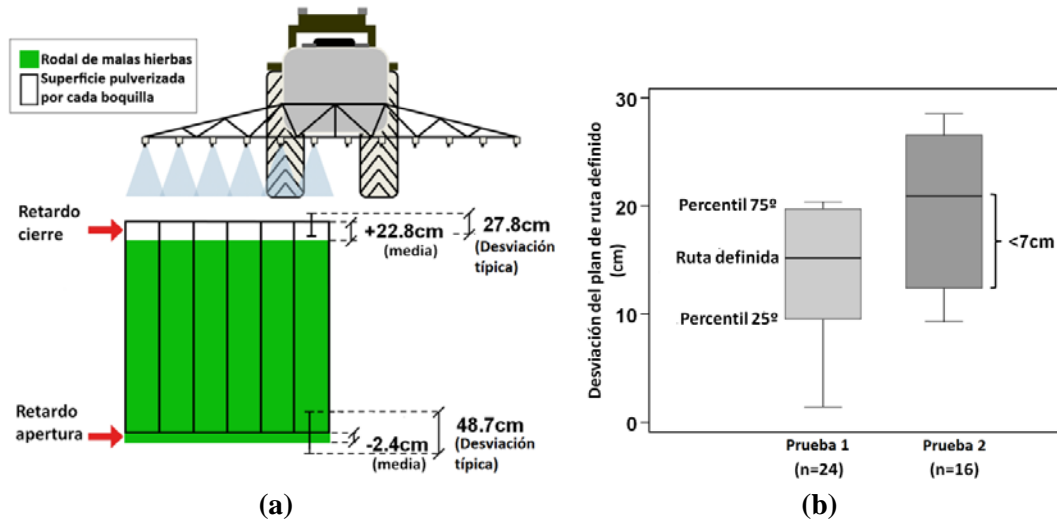


Figura 5.16. (a) Precisión en la apertura y cierre de las boquillas de pulverización de herbicida respecto al área objetivo. (b) Diferencias en las dos pruebas realizadas entre trayectoria real seguida por el vehículo y la trayectoria establecida.

En (Demostración Enero, 2014) hay disponible un vídeo en el que se muestran los momentos más interesantes de las dos jornadas de demostración del proyecto RHEA en las que funcionaron todos los sistemas presentados en este trabajo de tesis, a saber, el Planificador, el Supervisor y el Gestor. Además, en (Demostración Mayo, 2014) hay disponible otro vídeo en el que se muestra la ejecución automática de los otros dos escenarios del proyecto RHEA explicados al final de la sección 1.3, un tratamiento de eliminación de malas hierbas con detección en tiempo real en cultivos de calle ancha y un tratamiento de pulverización de fitosanitarios en cultivos leñosos. Ambos escenarios también se ejecutaron automáticamente utilizando el Planificador, Supervisor y Gestor de misiones propuestos en este trabajo de investigación.

Las contribuciones y resultados aportados por el Gestor de Misiones presentado en este capítulo se publicaron en (Ribeiro, y otros, 2015) y (Conesa-Muñoz, Valente, Del Cerro, Barrientos, & Ribeiro, 2015).

6 CONCLUSIONES Y TRABAJO FUTURO

6.1. Conclusiones

La principal contribución de esta tesis consiste en la automatización a alto nivel que se ha llevado a cabo para ejecutar tareas agrícolas de forma totalmente autónoma mediante flotas de robots. Esta automatización se fundamenta en las aportaciones que se han realizado en los campos de la planificación, supervisión y gestión de misiones agrícolas.

6.1.1. Planificación

El presente trabajo ha puesto de manifiesto que las trayectorias para llevar a cabo una tarea agrícola de forma eficiente varían en gran medida según el escenario agrícola en el que se trabaje. Para abordar este problema, se ha diseñado e implementado un planificador muy general, que ha sido capaz de encontrar las rutas a seguir para ejecutar de forma óptima tareas agrícolas en multitud de escenarios diferentes. A saber, usando uno o varios vehículos a la vez con características heterogéneas, con limitaciones físicas reales, tales como radio de giro o capacidad de almacenamiento (de combustible o fitosanitario), considerando la posibilidad de reabastecimiento, teniendo en cuenta la variabilidad del cultivo, trabajando con cualquier forma de campo e incluso con campos compuestos por varias parcelas, y, por último, optimizando para diferentes criterios, bien combinándolos adecuadamente en un único objetivo o mediante optimizaciones multi-objetivo.

El Planificador se ha desarrollado utilizando técnicas meta-heurísticas, más lentas que los habituales algoritmos voraces, pero ha demostrado obtener mejores soluciones en un tiempo razonable, del orden de minutos en el peor de los casos para los campos utilizados en las pruebas. En concreto se han probado tres algoritmos diferentes: Recocido Simulado, Algoritmo Genético y NSGA-II, cada uno de ellos con alguna propiedad particular interesante que lo diferencia del resto. Así, el Recocido Simulado es el más sencillo computacionalmente y utiliza una única solución para recorrer el espacio de búsqueda, mientras que el Algoritmo Genético y el NSGA-II emplean una población

de soluciones en cada iteración, siendo además el NSGA-II capaz de abordar la optimización de varios criterios simultáneamente.

Para mejorar la convergencia de los métodos de optimización, se ha propuesto un nuevo operador, mix-opt, responsable de la construcción de las nuevas soluciones en los métodos de optimización propuestos. El nuevo operador combina ponderadamente los operadores más utilizados en problemas de planificación de rutas y se ha comparado con los operadores que aparecen en la literatura, mostrando un mejor comportamiento tanto en problemas clásicos de rutas de vehículos como en problemas de planificación agrícola. Por tanto, se puede concluir que el operador propuesto en este trabajo de tesis es una contribución útil que permite mejorar el funcionamiento de los métodos de optimización meta-heurísticos en problemas de obtención de rutas mínimas.

Los tres métodos de optimización estudiados se han sometido a un proceso de configuración consistente en probar multitud de valores de configuración y quedarse con aquellos que producían mejores resultados en media. Gracias a este proceso se han podido comparar las tres técnicas en las mismas condiciones, es decir, con su mejor configuración, y determinar cuál es capaz de obtener mejores resultados. De los tres métodos, el Recocido Simulado ha demostrado ser el que mejores resultados obtiene para tiempos inferiores a 1h, mientras que el Algoritmo Genético es capaz de mejorar algo los resultados anteriores con tiempos de ejecución superiores.

Se ha propuesto un planificador basado en un algoritmo Recocido Simulado, una segmentación del campo en pistas, una representación de la solución con vectores de pistas y reabastecimientos, y un operador, mix-opt, como método de construcción de soluciones vecinas.

Se ha diseñado un amplio y variado repertorio de problemas que sintetizan muchos escenarios habituales en planificación agrícola, escenarios en muchos casos no considerados en los trabajos que se recogen en la literatura por abordar el problema de planificación de una forma más sencilla, y se ha comprobado que el Planificador propuesto es capaz de obtener soluciones buenas en todos los casos en tiempos inferiores a 1 minuto.

Gracias a este repertorio de problemas ilustrativos, se ha comprobado también que los reabastecimientos y la capacidad de los tanques de almacenamiento de los vehículos afectan a las trayectorias óptimas. Se ha demostrado que la variabilidad del campo puede influir en las trayectorias. Se ha comprobado que las trayectorias óptimas para flotas heterogéneas pueden ser diferentes a las de una flota homogénea, para aprovechar más eficientemente las características particulares de cada vehículo. Por último, se ha demostrado que planificar eficientemente una tarea agrícola no consiste únicamente en encontrar las trayectorias con una distancia mínima, como plantean la gran mayoría de los trabajos de la literatura. Se ha constatado que, cuando el problema se generaliza para abarcar otros costes también importantes como el tiempo, el combustible consumido o el coste por insumos, las trayectorias óptimas pueden resultar realmente diferentes de las rutas que minimizan la distancia.

Además de ser más general, se ha probado que el Planificador propuesto mejora los resultados de las aproximaciones formuladas con anterioridad por otros autores, empleando el Planificador para optimizar rutas en campos iguales a los considerados en estos estudios previos. En concreto, el Planificador superó los resultados presentados en (Bochtis & Vougioukas, 2008), disminuyendo la distancia recorrida en cabecera hasta en un 17 %, y los presentados en (Hameed, Bochtis, & Sorensen, 2011), con una reducción de un 21 % en distancia en cabecera.

A tenor de todos estos resultados, se puede concluir que el Planificador propuesto es una contribución muy valiosa por abordar y resolver el problema de planificación agrícola con una perspectiva generalista que incluye el uso de más de un vehículo en múltiples escenarios no considerados hasta el momento y que, además, ha demostrado generar mejores soluciones en los escenarios utilizados en trabajos previos.

6.1.2. Supervisión

Para controlar que la flota ejecuta la tarea según lo planificado, se ha diseñado, implementado y evaluado un sistema capaz de monitorizar una flota de vehículos autónomos detectando con eficacia el mal funcionamiento de los vehículos de la flota así como de los dispositivos de a bordo.

Para implementar este sistema, se ha propuesto una arquitectura distribuida jerárquica basada en los conceptos de supervisor y alarma. Estos dos conceptos han demostrado ser muy útiles porque permiten desacoplar la supervisión en módulos sencillos e independientes, muy fáciles de implementar. Además, como los supervisores pueden emplearse para monitorizar otros supervisores, es posible construir incrementalmente grandes sistemas globales de supervisión en varias capas, desarrollando primero niveles bajos dedicados a la monitorización de los comportamientos más sencillos, y después niveles más altos, fruto de la combinación de los primeros, dedicados al control de los comportamientos más complejos. Además los supervisores pueden ejecutarse fácilmente en varias máquinas diferentes, gracias a la distribución de proceso que proporciona la arquitectura propuesta.

La arquitectura ha demostrado ser de gran utilidad permitiendo construir con facilidad un sistema de supervisión distribuido sencillo pero efectivo, muy apropiado para vigilar flotas de vehículos autónomos por su naturaleza distribuida. Partiendo de esta arquitectura se diseñó un sistema Supervisor con tres niveles. El más bajo para operar dentro de las unidades y encargarse de los problemas más urgentes. Un nivel más alto para hacerse cargo de la supervisión más compleja, aquella que involucra a toda la flota, por lo que se ejecuta en un ordenador externo que recibe en tiempo real toda la información emitida por todas las unidades. Por tanto este nivel tiene una visión completa del estado de todos los elementos de la flota. Por último, un tercer nivel para permitir que un operario monitorice todo el sistema y tome control de los elementos de la flota en caso de ser necesario. Se ha comprobado que el sistema propuesto es capaz de implementar satisfactoriamente las tres funcionalidades de la supervisión, a saber, detección de fallos, diagnóstico y reparación, y que además puede incluir estas funcionalidades en cualquiera de sus tres niveles. Se comprobó diseñando dos pruebas en campos reales en las que unos tractores debían llevar a cabo una tarea agrícola real. En ambos casos el sistema de supervisión superó los test con un alto rendimiento, siendo capaz de detectar problemas en el estado del apero (apertura/cierre de boquillas con un pequeño desfase debidas al tiempo de respuesta del dispositivo), desviaciones en la velocidad de trabajo o riesgos de colisión. En este último caso, el sistema fue incluso capaz de contrarrestar el problema pausando uno de los vehículos para permitir el paso del otro, y, posteriormente, reanudar la marcha del vehículo parado una vez que ya no había riesgo de colisión.

Asimismo, se probó que el sistema permite realizar un seguimiento preciso de la misión a un único operario a cargo de la flota, ya que el sistema notificó puntual y detalladamente todo lo que sucedía, por lo que la situación no dejó de estar nunca bajo

control y, en caso de haber sido necesario, el operario podría haber tomado directamente el control de alguno de los vehículos de la flota.

Por último, se ha verificado mediante pruebas reales que el sistema implementado responde con fluidez en tiempo real y que admite con facilidad nuevos comportamientos de supervisión que detecten nuevas anomalías, simplemente incluyendo nuevos módulos supervisores en el nivel correspondiente y conectando las alarmas convenientemente.

Por todo ello, se puede concluir que el sistema de supervisión diseñado, implementado y evaluado es una valiosa herramienta para supervisar la ejecución de una flota de robots autónomos durante la realización de una tarea agrícola.

6.1.3. Gestión

Respecto a la gestión de la misión, se ha diseñado e implementado un sistema de gestión de misión que permite ejecutar de forma totalmente autónoma misiones agrícolas integrando de modo general el trabajo de robots aéreos y terrestres. El sistema está formado por módulos independientes que se encargan de implementar las etapas fundamentales necesarias para completar cualquier tipo de misión agrícola genérica utilizando flotas de robots. A saber, planificadores aéreos/terrestres, controladores de ejecución de la misión, supervisores de misión aérea/terrestre y módulos de procesamiento de datos. Mediante estos cuatro tipos de módulos se ha demostrado que es posible secuenciar de forma automática los pasos necesarios para implementar la gran mayoría de las tareas agrícolas habituales.

El sistema se ha integrado con el Planificador y Supervisor desarrollados en este trabajo de investigación, además de con un planificador y un supervisor para los vehículos aéreos, permitiendo así agrupar en un único sistema las etapas de planificación y supervisión para flotas aéreas y terrestres.

Se ha diseñado e implementado un módulo que permite controlar la ejecución de las tareas desde la perspectiva de flota considerando las órdenes habituales que pueden ejecutar los vehículos, tales como arrancar, pausar, reanudar y parar, para así poder controlar la ejecución de la misión. Este módulo, que se ha integrado en el sistema de gestión, es necesario aunque los vehículos empleados sean autónomos ya que la flota en sí no lo es, por lo que es necesario controlar los vehículos desde una perspectiva global para que trabajen coordinadamente como un equipo. Gracias a este módulo el sistema de gestión puede implementar la ejecución de la misión, por ejemplo, arrancando, pausando, reanudando, parando o enviando trayectorias a todos los miembros de la flota a la vez cuando la misión así lo requiere.

Por último, el sistema se ha diseñado para integrar fácilmente módulos que procesen y almacene la información adquirida durante la ejecución de una misión de tal modo que la información puede ser utilizada posteriormente en el diseño o ejecución de una nueva misión. De esta manera el sistema de gestión permite separar la ejecución de las tareas agrícolas en dos etapas; una primera de inspección, llevada a cabo por unidades aéreas con mayor capacidad para cubrir fácilmente grandes superficies y extraer información del cultivo, y una segunda de tratamiento, a cargo de vehículos terrestres, en la que ejecutar la tarea agrícola con precisión gracias al conocimiento adquirido en la primera etapa, por ejemplo, utilizando el mapa de distribución de las malas hierbas. Este planteamiento en dos etapas permite implementar con facilidad el ciclo de trabajo habitual, inspección/tratamiento, de la Agricultura de Precisión.

El sistema global se probó llevando a cabo una misión agrícola real, en concreto un tratamiento localizado de malas hierbas en cereal, compuesto por una misión aérea de inspección mediante drones, que determinaba la posición de los rodales de las malas hierbas en el cultivo, y una misión terrestre de tratamiento en la que un tractor iba pulverizando con precisión solo las zonas en las que había malas hierbas. La misión pudo ser realizada y supervisada por un único operario a través de una interfaz, excepto en los momentos puntuales de despegue y aterrizaje de las unidades aéreas, que tuvieron que ejecutarse manualmente con un piloto experimentado por exigencias de la legislación española vigente, y en la adquisición de las imágenes tomadas por los drones y que quedaban almacenadas en la tarjeta de memoria de las cámaras, ya que éstas no tenían una conexión directa al ordenador de a bordo que permitiese retransmitir las imágenes a la estación base. La misión se ejecutó además muy eficientemente, ya que se pulverizó el 97 % de las zonas objetivo (los rodales). A partir de los resultados obtenidos se puede concluir que el sistema de gestión de misiones propuesto es de enorme utilidad, por permitir integrar de modo efectivo las tareas de inspección realizadas por una flota de vehículos aéreos no tripulados con las de intervención realizadas por una flota de vehículos terrestres no tripulados y habilitar los medios para que un único operario pueda monitorizar con facilidad toda una flota de robots trabajando coordinadamente en un cultivo.

6.2. Trabajo futuro

Como trabajo futuro se proponen varias mejoras principalmente orientadas a conseguir un sistema global todavía más general, es decir, que sea válido en un conjunto mayor de escenarios agrícolas, con mejor rendimiento y que permita una ejecución todavía más automatizada, esto es, que soslaye aquellos pasos en los que por el momento es necesaria la intervención humana.

6.2.1. Planificación

En ocasiones los campos agrícolas presentan obstáculos tales como aspersores, pozos, postes eléctricos, estanques, etc. que limitan las trayectorias de los vehículos. En estos casos el Planificador tiene que tener en cuenta la posición de estos elementos para evitar que se produzcan colisiones y salvaguardar la seguridad de los vehículos en el campo. Por ejemplo, una posible estrategia puede consistir en penalizar el coste de las trayectorias que se vean interceptadas por los obstáculos para que estas soluciones no estén nunca entre las mejores. Otra posible estrategia es la de estimar pequeñas trayectorias iniciales seguras en aquellas zonas en las que están presentes los obstáculos y utilizar estas trayectorias en las zonas conflictivas. Por ejemplo, si hay un obstáculo en una cabecera, estimar las rutas de transición en esa cabecera mediante algún método capaz de evitar obstáculos en lugar de empleando las ecuaciones generales de Dubins.

Esta última solución es también la propuesta adecuada para resolver aquellos casos en los que, por la forma irregular del campo o por estar dividido en varias parcelas unidas por caminos preestablecidos, las ecuaciones de Dubins pueden no determinar trayectorias válidas para transitar entre las pistas (véase Figura 6.1). Empleando la estrategia de integrar en la solución rutas preestablecidas para algunas zonas específicas, el

planteamiento presentado en la tesis sigue siendo válido salvo que en lugar de estimar las transiciones utilizando el teorema de Dubins (Dubins, 1957) y los razonamientos geométricos de (Shkel & Lumelsky, 2001), se deberá utilizar algún otro método, como el descrito en (Wang, Soh, Wang, & Wang, 2002), que permita considerar las restricciones espaciales.

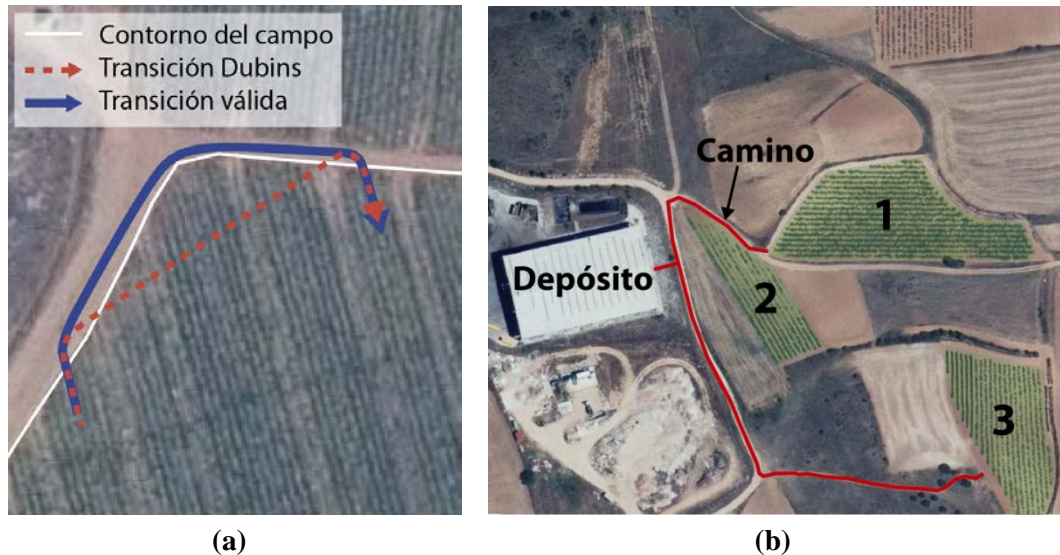


Figura 6.1. Transiciones válidas en (a) un campo irregular y en (b) un campo formado por varias parcelas conectadas por caminos.

Otra posible generalización es la de admitir vehículos con diferente ancho de trabajo, por ejemplo porque se dispongan de barras de pulverización de diferente anchura. Esta extensión no es nada sencilla porque afecta a la segmentación en pistas que se hace del campo. Si se consideran diferentes anchos de trabajo ya no está claro qué ancho de pista debe usarse para dividir el campo en partes disjuntas. Una posible solución puede consistir en emplear un ancho de pista igual al menor ancho de trabajo disponible, y seguir buscando el orden óptimo para recorrer las pistas. Sin embargo, debido a que ahora sí que hay aperos con un ancho de trabajo superior al de la pista, se pueden producir solapamientos y colisiones en las soluciones. Para resolver este problema, una estrategia es añadir una penalización en el coste de la solución en proporción al solapamiento que ésta produzca, de este modo el proceso de optimización convergerá hacia soluciones que minimicen esta situación.

En el trabajo (Hameed, Bochtis, & Sorensen, 2011) no se parte de un campo con un cultivo establecido y en este caso, la primera etapa consiste en determinar la mejor forma de establecer el cultivo encontrando una dirección que reduzca el número de transiciones necesarias para recorrer el campo. Esta primera etapa de determinación de la dirección óptima para establecer un cultivo puede ser una línea futura de desarrollo con la que completar aún más las funcionalidades del Planificador propuesto.

Por último, con la propuesta actual no es posible re-planificar en tiempo real en caso de que, por ejemplo, una unidad falle y las demás deban repartirse el trabajo que ésta tenía asignado y que queda por hacer. Re-planificar el trabajo restante es sencillo, basta

con determinar qué pistas se han tratado ya, y planificar de nuevo con las pistas restantes y considerando las unidades operativas. Sin embargo, como los fallos suceden mientras la misión se está ejecutando, la re-planificación debe poder hacerse en tiempo real. Esto implica que la planificación debe obtener buenos resultados más rápidamente. Téngase en cuenta que con la implementación actual, como se vio en la sección de resultados, si los campos son grandes pueden ser necesarios tiempos de ejecución del orden de decenas de minutos para alcanzar una solución razonablemente buena. Para acelerar la planificación se propone, como primera medida, implementar el Planificador en C o C++, para obtener con ello tiempos de computación de hasta 500 veces por debajo de los proporcionados por la plataforma Matlab (Andrews, 2012). Asimismo se podría mejorar la calidad de las soluciones y disminuir el tiempo de convergencia si el Planificador no parte de una población inicial generada aleatoriamente y parte de una población de cierta calidad generada mediante algún algoritmo voraz, por ejemplo el algoritmo de ahorros (Clarke & Wright, 1964) o el algoritmo 2-opt iterativo (Croes, 1958).

Otra opción para acelerar el Planificador es la de no recalcular totalmente los costes cada vez que se modifica una solución. Conociendo las modificaciones es posible calcular el nuevo coste de modo incremental a partir del antiguo aplicando las diferencias que suponen las modificaciones. Por ejemplo, si en una solución solo se ha modificado una transición, el nuevo coste puede obtenerse a partir del antiguo restando el coste para la antigua transición y sumando el de la nueva.

Por último, para que las planificaciones sean todavía más realistas, convendría considerar mapas de elevación del terreno para calcular con mayor precisión el consumo de combustible debido a la maquinaria siguiendo, por ejemplo, los modelos de consumo expuestos en (Hameed, 2014).

6.2.2. Supervisión

Como trabajo futuro para la supervisión, convendría aumentar la capacidad de reparación de fallos del sistema. Esto puede hacerse conectando más alarmas al Módulo Reparador de Fallos para que se disparen más protocolos de neutralización. Por ejemplo, variar la velocidad cuando se detecte que no es la apropiada, corregir la trayectoria o activar antes o después el aforo según el retardo que se esté detectando en la actuación.

Por otro lado, se ha comprobado que el sistema responde en tiempo real de forma fluida para flotas de hasta tres vehículos (el tamaño de la flota de RHEA), no obstante, deberían realizarse pruebas con un mayor número de vehículos para comprobar más exhaustivamente la escalabilidad del sistema. Flotas con más vehículos proporcionarían en tiempo real más información que procesar lo que podría saturar las comunicaciones y suponer un problema en la ejecución de todo el sistema de supervisión.

Asimismo, para evitar los casos en los que las alarmas se activen/desactiven con mucha frecuencia, puede resultar conveniente añadir un filtro al Gestor de Notificación de Alarmas para no abrumar al operario con información redundante y desviar su atención de otros eventos más importantes.

Por último, sería interesante que los protocolos de generación de alarmas, incluido la relevancia de la alarma, y así como de reparación, fuesen flexibles de modo que se pudiesen ajustar en tiempo de ejecución, bien manualmente por parte del operario o automáticamente por parte del sistema, dependiendo de la carga que tenga el operario en cada momento.

6.2.3. Gestión

Respecto a la gestión, como trabajo futuro se propone evitar el paso manual que supone llevar las tarjetas con las fotografías adquiridas por los drones hasta el ordenador de la estación base para el volcado de las imágenes, e integrar una etapa de volcado automático.

Asimismo, es necesario probar el sistema con todo tipo de tareas agrícolas (arado/laboreo, siembra, inspección fertilización, cosecha, etc.) para comprobar que un único operario puede supervisar el trabajo realizado por varios vehículos de una flota, incluida la integración de la tarea de inspección (robots aéreos), la información que ésta genera y la tarea de intervención (robots terrestres).

Por último, se estima conveniente completar el sistema para que la información, sobre todo las alarmas, llegue al operario de forma efectiva, añadiendo por ejemplo avisos sonoros. El problema más grave de la visualización de la información es la complejidad de los datos en el sentido del número de dimensiones que hay que representar simultáneamente en un espacio reducido de dos dimensiones, como es una pantalla. El problema que surge en este punto es el de cómo se puede reducir la complejidad de la información a mostrar a un grado que no complique la interpretación que el operario tiene que realizar para tomar una decisión adecuada en un corto intervalo de tiempo. Esto cobra especial importancia en aquellos casos en los que hay riesgo de accidente.

7 BIBLIOGRAFÍA

- ACK. (2015). Recuperado el Julio de 2015, de Wikipedia: <https://es.wikipedia.org/wiki/ACK>
- Affenzeller, M., Wagner, S., Winkler, S., & Beham, A. (2009). *Genetic algorithms and genetic programming: modern concepts and practical applications*. Crc Press.
- Agriculture in the European Union - Markets Statistical Information 2014. (Diciembre de 2014). Recuperado el Julio de 2015, de Agriculture and Rural Development European Commission: http://ec.europa.eu/agriculture/markets-and-prices/market-statistics/pdf/2014/full-report_en.pdf
- AirRobot Company. (2015). Recuperado el Septiembre de 2015, de <http://www.airrobot.de/>
- Andrews, T. (2012). Computation Time Comparison Between Matlab and C++ Using Launch Windows. *Aerospatial Engineering* (78), 1-6.
- Andújar, D., Ribeiro, A., Fernández-Quintanilla, C., & Dorado, J. (2011). Reliability of a Visual Recognition System for Detection of Johnsongrass (*Sorghum halepense*) in Corn. *Weed Technology* , 25 (4), 645-651.
- Applegate, D. L., Bixby, R. E., Chvatal, V., & Cook, W. J. (2006). *The Traveling Salesman Problem: A Computational Study*. Princeton: Princeton university press.
- Autonomous Tractor Corporation. (Julio de 2015). *Spirit Tractor*. Obtenido de Autonomous Tractor Corporation Website http://www.autonomoustractor.com/INSIDES/the_spirit.html
- Bak, T., & Jakobsen, H. (2004). Agricultural robotic platform with four wheel steering for weed detection. *Biosystems Engineering* , 87 (2), 125-136.
- Baldacci, R., Battarra, M., & Vigo, D. (2008). Routing a heterogeneous fleet of vehicles. En *The vehicle routing problem: latest advances and new challenges* (págs. 3-27). Springer.
- Banzhaf, W. (1990). The “molecular” traveling salesman. *Biological Cybernetics* , 64 (1), 7-14.
- Barawid, O. C., Mizushima, A., Ishii, K., & Noguchi, N. (2007). Development of an autonomous navigation system using a two-dimensional laser scanner in an orchard application. *Biosystems Engineering* , 96 (2), 139-149.
- Barrientos, A., Colorado, J., Cerro, J., Martinez, A., Rossi, C., Sanz, D., y otros. (2011). Aerial remote sensing in agriculture: A practical approach to area coverage and

- path planning for fleets of mini aerial robots. *Journal of Field Robotics* , 28 (5), 667-689.
- Bengochea-Guevara, J. M., Conesa-Muñoz, J., & Ribeiro, A. (2012). Developing a small autonomous robot for crop inspection. *Proceedings of the International Conference of Agricultural Engineering-CIGR-AgEng 2012: Agriculture and Engineering for a Healthier Life*, (págs. C-2171). Valencia, Spain.
- Benson, E. R., Reid, J. F., & Zhang, Q. (2003). Machine vision-based guidance system for an agricultural small-grain harvester. *Transactions of the ASAE* , 46 (4), 1255–1264.
- Berny, P. (2007). Pesticides and the intoxication of wild animals. *Journal of veterinary pharmacology and therapeutics* , 30 (2), 93-100.
- Blum, C., & Roli, A. (2003). Metaheuristics in combinatorial optimization: Overview and conceptual comparison. *ACM Computing Surveys (CSUR)* , 35 (3), 268-308.
- Bochtis, D. D., & Sørensen, C. G. (2009). The vehicle routing problem in field logistics part I. *Biosystems Engineering* , 104 (4), 447-457.
- Bochtis, D. D., & Sørensen, C. G. (2010). The vehicle routing problem in field logistics: Part II. *Biosystems Engineering* , 105 (2), 180-188.
- Bochtis, D. D., & Vougioukas, S. G. (2008). Minimising the non-working distance travelled by machines operating in a headland field pattern. *Biosystems engineering* , 101 (1), 1-12.
- Bochtis, D. D., Sørensen, C. G., & Vougioukas, S. G. (2010). Path planning for in-field navigation-aiding of service units. *Computers and Electronics in Agriculture* , 74 (1), 80-90.
- Bochtis, D. D., Sørensen, C. G., Green, O., Moshou, D., & Olesen, J. (2010). Effect of controlled traffic on field efficiency. *Biosystems engineering* , 106 (1), 14-25.
- Bochtis, D., Vougioukas, S., Tsatsarelis, C., & Ampatzidis, Y. (2007). Optimal dynamic motion sequence generation for multiple harvesters. *Agricultural Engineering International: the CIGR ejournal* , 9 (1).
- Boomer 3000 series. (2015). Recuperado el Abril de 2015, de Case New Holland Website: http://agriculture.newholland.com/us/en/Products/Compact-Tractors-and-Commercial-Mowers/Boomer-3000-Series/Pages/3050_modelversion.aspx
- Brandstotter, M., Hofbaur, M. W., Steinbauer, G., & Wotawa, F. (2007). Model-based fault diagnosis and reconfiguration of robot drives. *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, (págs. 1203 - 1209). San Diego, CA.
- Burgos Artizzu, X. P. (2009). *Percepción y Actuación para el tratamiento selectivo de malas hierbas en tiempo real*. Madrid: Universidad Complutense de Madrid.
- Burgos-Artizzu, X. P., Ribeiro, A., Guijarro, M., & Pajares, G. (2011). Real-time image processing for crop/weed discrimination in maize fields. *Computers and Electronics in Agriculture* , 75 (2), 337-346.

- Cahn, M. D., Hummel, J. W., & Brouer, B. H. (1994). Spatial analysis of soil fertility for site-specific crop management. *Soil Science Society of America Journal* , 58 (4), 1240-1248.
- Carballido, J., Perez-Ruiz, M., Gliever, C., & Agüera, J. (2012). Design, development and lab evaluation of a weed control sprayer to be used in robotic systems. *Proceedings of the First International Conference on Robotic and Associated High-Technologies and Equipment for Agriculture, 1*, págs. 23-29. Pisa.
- Casbeer, D. W., Beard, R. W., McLain, T. W., Li, S. M., & Mehra, R. K. (2005). Forest fire monitoring with multiple small UAVs. *Proceedings of American Control Conference 2005, 5*, págs. 3530 -3535.
- Cerny, V. (1985). Thermodynamical approach to the traveling salesman problem: An efficient simulation algorithm. *Optimization Theory and Applications* (45), 41–51.
- Chateau, T., Debain, C., Collange, F., Trassoudaine, L., & Alizon, J. (2000). Automatic guidance of agricultural vehicles using a laser sensor. *Computers and electronics in agriculture* , 28 (3), 243-257.
- Chiang, L. H., Braatz, R. D., & Russell, E. L. (2001). *Fault Detection and Diagnosis in Industrial Systems*. London, Great Britain: Springer.
- Choset, H., & Pignon, P. (1998). Coverage path planning: The boustrophedon cellular decomposition. *Field and Service Robotics* , 203-209.
- Christofides & Eilon's Instances*. (March de 2015). Obtenido de <http://neo.lcc.uma.es/vrp/vrp-instances/capacitated-vrp-instances/>
- Clarke, G., & Wright, J. (1964). Scheduling of vehicles from a central depot to a number of delivery points. *Operations Research* , 12 (4), 568-581.
- COAG. (2015). *Evolución de los costes de producción 2014 / Influencia petróleo y gas*. http://www.coag.org/rep_ficheros_web/5e4da87675a96c8b2e8eee9b0a3fe2d5.pdf.
- Conesa-Munoz, J., Bengochea-Guevara, J. M., Andujar, D., & Ribeiro, A. (2015). Efficient Distribution of a Fleet of Heterogeneous Vehicles in Agriculture: A Practical Approach to Multi-path Planning. *Proceedings of 2015 IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC)*, (págs. 56-61). Vila-Real.
- Conesa-Muñoz, J., & Ribeiro, A. (2015). *Collision Supervisor Test*. Obtenido de <https://www.youtube.com/watch?v=XeMrMdKyGiY>
- Conesa-Muñoz, J., & Ribeiro, A. (2012). Consolidation of a WSN and Minimax Method to Rapidly Neutralise Intruders in Strategic Installations. *Sensors* , 12 (3), 3281-3301.
- Conesa-Muñoz, J., & Ribeiro, A. (2015). *Spraying Mission Test*. Obtenido de <https://www.youtube.com/watch?v=FfDHA-JmfrC>
- Conesa-Muñoz, J., Gonzalez-de-Soto, M., Gonzalez-de-Santos, P., & Ribeiro, A. (2015). Distributed multi-level supervision to effectively monitor the operations of a fleet of autonomous vehicles in agricultural tasks. *Sensors* , 15 (3), 5402-5428.

- Conesa-Muñoz, J., Ribeiro, A., Andujar, D., Fernandez-Quintanilla, C., & Dorado, J. (2012). Multi-path planning based on a NSGA-II for a fleet of robots to work on agricultural tasks. *Proceedings of 2012 IEEE Congress on the Evolutionary Computation (CEC)* (págs. 1-8). Brisbane, Australia: IEEE.
- Conesa-Muñoz, J., Valente, J., Del Cerro, J., Barrientos, A., & Ribeiro, A. (2015). Integrating autonomous aerial scouting with autonomous ground actuation to reduce chemical pollution on crop soil. *Proceedings of ROBOT2015: Second Iberian Robotics Conference*. Lisboa: Springer International Publishing.
- Cong, J., Fang, J., & Khoo, K. Y. (2000). DUNE: a multi-layer gridless routing system with wire planning. *Proceedings of ISPD '00 Proceedings of the 2000 international symposium on Physical design* (págs. 12-18). ACM.
- Croes, G. A. (1958). A Method for Solving Traveling-Salesman Problems. *Operations Research*, 6, 791-812.
- Dantzig, G. B., & Ramser, J. H. (1959). The Truck Dispatching Problem. *Management Science*, 6 (80), 80-91.
- Darwin, C. (1859). *The origin of species by means of natural selection*. (Murray, Ed.) Londres, U.K.
- De Jong, K. A. (2006). *Evolutionary computation: a unified approach*. MIT press.
- Dearden, R., Willeke, T., Simmons, R., Verma, V., Hutter, F., & Thrun, S. (2004). Real-time fault detection and situational awareness for rovers: Report on the mars technology program task. *Proceedings of IEEE Aerospace Conference*, 2, págs. 826-840.
- Deb, K. (2014). Multi-objective Optimization. En *Search Methodologies* (págs. 403-449). Springer .
- Deb, K., Agrawal, S., Pratap, A., & Meyarivan, T. (2000). A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II. *Lecture notes in computer science*, (1917), 849-858.
- Del Cerro, J., Sanz, D., Valente, J., Rossi, C., Cancar, L., & Barrientos, A. (2014). Flota de vehículos aéreos para fotografía de alta resolución en aplicaciones de agricultura de precisión. *Actas de las XXXV Jornadas de Automática*. Valencia.
- Demo RHEA - Enero 2014. (Enero de 2014). Recuperado el 1 de Julio de 2015, de Proyecto RHEA:
http://www.rhea-project.eu/img/Videos/DEMO_short_version_01.mp4
- Demo RHEA - Mayo 2014. (Mayo de 2014). Recuperado el Julio de 2015, de Proyecto RHEA: <http://www.rhea-project.eu/img/Videos/RHEA.mp4>
- Demyen, D., & Buro, M. (2006). Efficient triangulation-based pathfinding. *Proceedings of the 21st National conference on Artificial intelligence (AAAI-06)*, (págs. 942-947). Boston.

- Dobermann, A., Witt, C., Dawe, D., Abdulrachman, S., Gines, H. C., Nagarajan, R., y otros. (2002). Site-specific nutrient management for intensive rice cropping systems in Asia. *Field Crops Research* , 74 (1), 37-66.
- Doering, C. (23 de March de 2014). Growing use of drones poised to transform agriculture. *USA Today* .
- Dréo, J. (2015). *Clasificación de metaheurísticas* (<http://es.wikipedia.org/wiki/Metaheurística>). Recuperado el 26 de Febrero de 2015, de http://commons.wikimedia.org/wiki/File:Metaheurísticas_clasificación.png
- Dubins, L. E. (1957). On curves of minimal length with a constraint on average curvature, and with prescribed initial and terminal positions and tangents. *American Journal of mathematics* , 497-516.
- ECPA. (2012). *Report Annual Review 2011*. Obtenido de European Crop Protection Association: http://www.ecpa.eu/files/attachments/Annual_Report_web.pdf
- Edan, Y. H. (2009). Automation in Agriculture. En *Springer Handbook of Automation* (págs. 1095-1128). Springer Berlin Heidelberg.
- Emmi, L. (2014). *Contribuciones a la configuración de flotas de robots para agricultura de precisión* (Tesis Doctoral ed.). Madrid.
- Emmi, L., Gonzalez-de-Soto, M., Pajares, G., & Gonzalez-de-Santos, P. (2014). Integrating Sensory/Actuation Systems in Agricultural Vehicles. *Sensors* , 14, 4014–4049.
- EUROSTAT. (2013). Recuperado el Agosto de 2015, de Pesticide sales (from 2009 onwards) - Reg. 1185/2009 (aei_fm_salpest09): http://appsso.eurostat.ec.europa.eu/nui/show.do?dataset=aei_fm_salpest09&lang=en
- Farm Works™ Mapping. (2015). Recuperado el Julio de 2015, de FarmWorks Software: <http://www.farmworks.com/products/mapping>
- FarmLogs. (Julio de 2015). Obtenido de <https://farmlogs.com/>
- Fendt GuideConnect. (2011). Recuperado el Julio de 2015, de Fendt: http://www.fendt.com/int/press_pressreleases_5636.asp
- Fishel, F., Bailey, W., Boyd, M., Johnson, B., O'Day, M., Sweets, L., y otros. (2001). *Introduction to Crop Scouting*. University of Missouri .
- Fogel, D. B. (1993). Applying evolutionary programming to selected traveling salesman problems. *Cybernetics and systems* , 24 (1), 27-36.
- García-Pérez, L., García-Alegre, M. C., Ribeiro, A., & Guinea, D. (2008). An agent of behaviour architecture for unmanned control of a farming vehicle. *Computers and Electronics in Agriculture* , 60 (1), 39–48.
- Gaudin, S. (2010 de Agosto de 2010). *MIT builds swimming, oil-eating robots*. Recuperado el Septiembre de 2015, de Computerworld: <http://www.computerworld.com/article/2514966/emerging-technology/mit-builds-swimming--oil-eating-robots.html>

- Gerhards, R., Sökefeld, M., Schulze-Lohne, K., Mortensen, D. A., & Kühbauch, W. (1997). Site specific weed control in winter wheat. *Journal of Agronomy and Crop Science* , 178 (4), 219-225.
- Gertler, J. (1998). *Fault detection and diagnosis in engineering systems*. New York: CRC press.
- Global Harvest Initiative. (2014). *2014 Global Agricultural Productivity Report*. Global Harvest Initiative.
- Goldberg, D. E., & Holland, J. H. (1988). Genetic algorithms and machine learning. *Machine learning* , 3 (2), 95-99.
- Goldberg, D. E., & Lingle, R. (1985). Alleles, loci, and the traveling salesman problem. *Proceedings of the first international conference on genetic algorithms and their applications* (págs. 154-159). Lawrence Erlbaum Associates, Publishers.
- Gomez-Gil, J., Alonso-Garcia, S., Gómez-Gil, F. J., & Stombaugh, T. (2011). A simple method to improve autonomous GPS positioning for tractors. *Sensors* , 11, 5630–5644.
- Gonzalez-de-Soto, M., Emmi, L., Garcia, I., & Gonzalez-de-Santos, P. (2015). Reducing fuel consumption in weed and pest control using robotic tractors. *Computers and Electronics in Agriculture* , 114, 96-113.
- Gottschalk, R., Burgos-Artizzu, X. P., Ribeiro, A., Pajares, G., & Miralles, A. S. (2009). Real-time image processing for the guidance of a small agricultural field inspection vehicle. *International journal of intelligent systems technologies and applications* , 8 (1-4), 434-443.
- Guerrero, J. M., Guijarro, M., Montalvo, M., Romeo, J., Emmi, L., Ribeiro, A., y otros. (2013). Automatic expert system based on images for accuracy crop row detection in maize fields. *Expert Systems with Applications* , 40 (2), 656–664.
- Gutin, G., & Punnen, A. (2002). *The Traveling Salesman Problem and Its Variations*. Springer Science & Business Media.
- Hague, T., Marchant, J. A., & Tillett, N. D. (2000). Ground based sensing systems for autonomous agricultural vehicles. *Computers and Electronics in Agriculture* , 25 (1-2), 11–28.
- Hameed, I. A. (2014). Intelligent Coverage Path Planning for Agricultural Robots and Autonomous Machines on Three-Dimensional Terrai. *Journal of Intelligent & Robotic Systems* , 74 (3-4), 965-983.
- Hameed, I. A., Bochtis, D. D., & Sorensen, C. G. (2011). Driving angle and track sequence optimization for operational path planning using genetic algorithms. *Applied Engineering in Agriculture* , 27 (6), 1077-1086.
- Hamza, M., & Anderson, W. K. (2005). Soil compaction in cropping systems: A review of the nature, causes and possible solutions. *Soil and Tillage Research* , 82 (2), 121-145.

- Han, S., Zhang, Q., Ni, B., & Reid, J. F. A guidance directrix approach to vision-based vehicle guidance systems. *Computers and Electronics in Agriculture* , 43 (3), 179-195.
- Harel, D. (1987). Statecharts: A visual formalism for complex systems. *Science of computer programming* , 8 (3), 231-274.
- Harel, D., & Pnueli, A. (1985). *On the development of reactive systems*. Berlin: Springer Berlin Heidelberg.
- Holland, J. H. (1975). *Adaptation in natural and artificial systems: An introductory analysis with applications to biology, control, and artificial intelligence*. Michigan Press.
- Ingber, L. (1989). Very fast simulated re-annealing. *Mathematical and computer modelling* , 12 (8), 967-973.
- Jensen, M. A., Bochtis, D., Sørensen, C. G., Blas, M. R., & Lykkegaard, K. L. (2012). In-field and inter-field path planning for agricultural transport units. *Computers & Industrial Engineering* , 63 (4), 1054-1061.
- John Deere Guidance Systems*. (2015). Recuperado el Julio de 2015, de John Deere Website:
https://m.deere.com/tr_TR/docs/product/equipment/agricultural_managment_solutions/guidance_systems/autotrac_rowsense/yy1114823_e.pdf
- Kallmann, M., Bieri, H., & Thalmann, D. (2004). Fully Dynamic Constrained Delaunay Triangulations. En G. Brunnert, B. Hamann, H. Müller, & L. Linsen (Edits.), *Geometric modeling for scientific visualization* (págs. 241-257). Berlin: Springer Berlin Heidelberg.
- Kirkpatrick, S., Gelatt, C. D., & Vecchi, M. P. (1983). Optimization by simulated annealing. *Science* , 220 (4598), 671-680.
- Larrañaga, P., Kuijpers, C. M., Murga, R. H., Inza, I., & Dizdarevic, S. (1999). Genetic algorithms for the travelling salesman problem: A review of representations and operators. *Artificial Intelligence Review* , 13 (2), 129-170.
- Lee, Y. H., Kim, J. I., Kang, K. H., & Kim, K. H. (2008). A Heuristic for Vehicle Fleet Mix Problem Using Tabu Search and Set Partitioning. *The Journal of the Operational Research Society* , 59 (6), 833-841.
- Li, M., Imou, K., Wakabayashi, K., & Yokoyama, S. (2009). Review of research on agricultural vehicle autonomous guidance. *International Journal of Agricultural and Biological Engineering* , 2 (3), 1-16.
- Lin, S. (1965). Computer solutions of the traveling salesman problem. *The Bell System Technical Journal* , 44 (10), 2245-2269.
- Márquez, L. (2007). Los tractores en la agricultura española. Parte 2. *Agrotécnica* , 68-73.
- McCarthy, C. E., & Pollack, M. E. (2000). Towards focused plan monitoring: A technique and an application to mobile robots. *Autonomous Robots* , 9 (1), 71-81.

- MCPA 60 SL Herbicida. (2015). Recuperado el Julio de 2015, de Dupont: <http://www.dupont.es/productos-y-servicios/proteccion-de-cultivos/cultivo-de-arroz-cultivo-de-cereales/productos/mcpa-60-sl-malas-hierbas-en-arroz.html>
- Michalewicz, Z. (1996). *Genetic algorithms+ data structures= evolution programs*. Springer Science & Business Media.
- Michie, D. (1968). "Memo" Functions and Machine Learning. *Nature* (218), 19 - 22.
- Miller, B. L., & Goldberg, D. E. (1995). Genetic algorithms, tournament selection, and the effects of noise. *Complex Systems* , 9 (3), 193-212.
- Miller, G. T. (2004). *Sustaining the Earth: An Integrated Approach*. Brooks/Cole.
- Mohammaditabar, D. (2013). Chinese Postman Problem. En R. Z. Farahani, & E. Miandoabchi, *Graph Theory for Operations Research and Management: Applications in Industrial Engineering: Applications in Industrial Engineering* (págs. 224-228). IGI Global.
- Montalvo, M., Guerrero, J., Romeo, J., Emmi, L., Guijarro, M., & Pajares, G. (2013). Automatic expert system for weeds/crops identification in images from maize fields. *Expert Systems with Applications* , 40, 75-82.
- Mousazadeh, H. (2013). A technical review on navigation systems of agricultural autonomous off-road vehicles. *Journal of Terramechanics* , 50 (3), 211-232.
- Mulla, D. J. (2013). Twenty five years of remote sensing in precision agriculture: Key advances and remaining knowledge gaps. *Biosystems Engineering* , 114 (4), 358-371.
- Nathan, M., Shaojie, S., Kartik, M., Yash, M., Vijay, K., Keiji, N., y otros. (2012). Collaborative mapping of an earthquake-damaged building via ground and aerial robots. *Journal of Field Robotics* , 29 (5), 832-841.
- Navas, L. M., González, F., Gómez, J., Ruiz, G., Barreiro, P., D, A., y otros. (2010). *Ahorro y Eficiencia Energética en la Agricultura de Precisión*. Madrid: IDAE.
- Oerke, E. C., Dehne, H. W., Schönbeck, F., & Weber, A. (1994). *Crop Production and Crop Protection. Estimated losses in major food and cash crops*. Amsterdam, The Netherlands: Elsevier.
- Oksanen, T., & Visala, A. (2009). Coverage path planning algorithms for agricultural field machines. *Journal of Field Robotics* , 26 (8), 651-668.
- Oliver, I. M., Smith, D., & Holland, J. R. (1987). Study of permutation crossover operators on the traveling salesman problem. *Genetic algorithms and their applications: proceedings of the second International Conference on Genetic Algorithms*.
- Oliver, M. A. (2010). *Geostatistical Applications for Precision Agriculture*. (M. A. Oliver, Ed.) Dordrecht: Springer.
- Optimizer. (2015). Recuperado el Julio de 2015, de Advance AG Solutions: <http://www.advancedagsolutions.com/optimizer.php#opt2>

- Or, I. (1976). *Traveling salesman-type combinatorial problems and their relation to the logistics of regional blood banking*. Xerox University Microfilms.
- Pardalos, P. M., Migdalas, A., & Pitsoulis, L. (Edits.). (2008). *Pareto optimality, game theory and equilibria*. Springer Science & Business Medi.
- Pedersen, S. M., Fountas, S., Have, H., & Blackmore, B. S. (2006). Agricultural robots—system analysis and economic feasibility. *Precision agriculture* , 74 (4), 295-308.
- Peña-Barragán, J. M., Torres-Sánchez, J., de Castro, A. I., Kelly, M., & López-Granados, F. (2013). Weed mapping in early-season maize fields using object-based analysis of unmanned aerial vehicle (UAV) images. *PLoS ONE* , 8-10.
- Perez-Ruiz, M., Carballido, J., Agüera, J., & Rodríguez-Lizana, A. (2013). Development and evaluation of a combined cultivator and band sprayer with a row-centering RTK-GPS guidance system. *Sensors* , 13, 3313–3330.
- Peruzzi, A., Frascioni, C., Martelloni, L., Fontanelli, M., & Raffaelli, M. (2012). Application of precision flaming to maize and garlic in the RHEA project. *Proceedings of the First International Conference on Robotics and Associated High-Technologies and Equipment for Agriculture, 1*, págs. 55–60. Pisa.
- Pettersson, O., Karlsson, L., & Saffiotti, A. (2005). Model-free execution monitoring by learning from simulation. *Proceedings on Computational Intelligence in Robotics and Automation*, (págs. 505-511).
- Pettersson, O., Karlsson, L., & Saffiotti, A. (2004). *Model-Free Execution Monitoring in Behavior-Based Mobile Robotic*. Örebro : Örebro universitetsbibliotek.
- Pierce, F. J., & Nowak, P. (1999). Aspects of Precision Agriculture. *Advances in agronomy* , 67, 1-85.
- Pimentel, D., Acquay, H., Biltonen, M., Rice, P., Silva, M., Nelson, J., y otros. (1992). Environmental and economic costs of pesticide use. *BioScience* , 42, 750-758.
- Prügel-Bennett, A. (2001). The mixing rate of different crossover operators. En W. Martin, & W. Spears (Edits.), *Foundations of genetic algorithms* (págs. 261-274).
- Pulverizador remolcado serie M700/M700i John Deere*. (s.f.). Recuperado el Julio de 2015, de John Deere Website:
https://www.deere.es/es_ES/products/equipment/sprayers/trailed_sprayers/m700_m700i_trailed_sprayers/m700_m700i_trailed_sprayers.page?
- Qt Libraries*. (2015). Recuperado el Julio de 2015, de Qt: <http://www.qt.io/developers/>
- Rabatel, G., & Labbé, S. (2015). A fully automatized processing chain for high-resolution multispectral image acquisition of crop parcels by UAV. *Proceedings of the 10th European Conference on Precision agriculture*, (págs. 135-141). Tel Aviv.
- Raffaelli, M., Martelloni, L., Frascioni, C., Fontanelli, M., & Peruzzi, A. (2013). Development of Machines for Flaming Weed Control on Hard Surfaces. *Applied Engineering in Agriculture* , 29 (5), 663-673.
- Ranganathan, N., Patel, M., & Sathyamurthy, R. (2001). An intelligent system for failure detection and control in an autonomous underwater vehicle. *Systems, Man and*

- Cybernetics, Part A: Systems and Humans, IEEE Transactions on* , 31 (6), 762 - 767.
- Raynal, M. (2013). Distributed Deadlock Detection. En *Distributed Algorithms for Message-Passing Systems* (págs. 401-423). Springer Berlin Heidelberg.
- Reid, J. F. (2011). The impact of mechanization on agriculture. En *The Bridge* (Vol. 41, págs. 22-29).
- RHEA. (2010-2014). Recuperado el 1 de Julio de 2015, de Proyecto RHEA Website: <http://www.rhea-project.eu/>
- Ribeiro, A., Fernandez-Quintanilla, C., Dorado, J., López-Granados, F., Manuel Peña, J., Rabatel, G., y otros. (2015). A fleet of aerial and ground robots: a scalable approach for autonomous site-specific herbicide application. *Precision agriculture'15* (págs. 153-159). Wageningen Academic Publishers.
- Rider, T. W., Vogel, J. W., Dille, J. A., Dhuyvetter, K. C., & Kastens, T. L. (2006). An economic evaluation of site-specific herbicide application. *Precision Agriculture* , 379-392.
- Robotnik. (2015). *Robotnik Agricultural Platforms*. Recuperado el Mayo de 2015, de Robotnik Website: <http://www.robotnik.eu/services-robotic/mobile-robotics-applications/agriculture/>
- Romeo, J., Guerrero, J., Montalvo, M., Emmi, L., Guijarro, M., Gonzalez-de-Santos, P., y otros. (2013). Camera sensor arrangement for crop/weed detection accuracy in agronomic images. *Sensors* , 13, 4348–4366.
- Ruckelshausen, A., Biber, P., Dorna, M., Gremmes, H., Klose, R., Linz, A., y otros. (2009). BoniRob—an autonomous field robot platform for individual plant phenotyping. En *Precision Agriculture '09* (págs. 841-847). Wageningen Academic Pub.
- Ruiz, D., Escribano, C., & Fernández-Quintanilla, C. (2006). Assessing the opportunity for site-specific management of *Avena sterilis* in winter barley fields in Spain. *Weed Research* , 46 (5), 379-387.
- Ryerson, A. E., & Zhang, Q. (2007). Vehicle path planning for complete field coverage using genetic algorithms. *Agricultural Engineering International: CIGR Journal* , IX.
- Saffiotti, A. (1999). Handling Uncertainty in Control of Autonomous Robots. En *Artificial Intelligence Today: Recent Trends and Developments* (págs. 381-407). Springer Berlin Heidelberg.
- Salhi, S., Imran, A., & Wassan, N. A. (2014). The multi-depot vehicle routing problem with heterogeneous vehicle fleet: Formulation and a variable neighborhood search implementation. *Computers & Operations Research* , 52 (Part B), 315-325.
- Shkel, A. M., & Lumelsky, V. (2001). Classification of the Dubins set. *Robotics and Autonomous Systems* , 179-202.

- Simmons, R., Fernandez, J. L., Goodwin, R., Koenig, S., & O'Sullivan, J. (2000). Lessons learned from Xavier. *IEEE Robotics and Automation Magazine* , 7 (2), 33-39.
- Sistemas agrícolas avanzados CIH*. (Julio de 2015). Obtenido de CIH Website: <http://assets.cnhindustrial.com/caseih/emea/EMEAASSETS/Products/AFS%C2%AE-Advanced-Farming-Systems/Brochures/13c8003eoo.pdf>
- Slaughter, D. C., Giles, D. K., & Downey, D. (2008). Autonomous robotic weed control systems: A review. *Computers and Electronics in Agriculture* , 61 (1), 63-78.
- Sørensen, C. G., & Bochtis, D. D. (2010). Conceptual model of fleet management in agriculture. *Biosystems Engineering* , 105 (1), 41-50.
- Sørensen, C. G., & Nielsen, V. (2005). Operational analyses and model comparison of machinery systems for reduced tillage. *Biosystems engineering* , 92 (2), 143-155.
- Sorensen, C. G., Bak, T., & Jorgensen, R. N. (2004). Mission planner for agricultural robotics. *AgEng 2004*. Leuven, Belgium.
- Stafford, J. V., Ambler, B., Lark, R. M., & Catt, J. (1996). Mapping and interpreting the yield variation in cereal crops. *Computers and Electronics in Agriculture* , 14 (2), 101-119.
- Stoll, A., Stafford, J., & Werner, A. (2003). Automatic operation planning for GPS-guided machinery. *Proceedings of 4th European Conference on Precision Agriculture* (págs. 657-664). Berlin, Germany: Wageningen Academic Publishers.
- Syswerda, G. (1991). *Handbook of genetic algorithms*. (L. Davis, Ed.) Van Nostrand Reinhold.
- Taïx, M., Souères, P., Frayssinet, H., & Cordesses, L. (2006). Path Planning for Complete Coverage with Agricultural Machines. *Field and Service Robotics* , 24, 549-558.
- The State Machine Compiler*. (2015). Obtenido de <http://smc.sourceforge.net/>
- Tillett, N. D., Hague, T., Grundy, A. C., & Dedousis, A. P. (2008). Mechanical within-row weed control for transplanted crops using computer vision. *Biosystems Engineering* , 99 (2), 171-178.
- Timmermann, C., Gerhards, R., & Kühbauch, W. (2003). The Economic Impact of Site-Specific Weed Control. *Precision Agriculture* , 4 (3), 249-260.
- Tombropoulos, R., Adlerb, J., & Latombec, J. (1999). CARABEAMER: a treatment planner for a robotic radiosurgical system with general kinematics. *Medical Image Analysis* , 3 (3), 237-264.
- Topcon Precision Agriculture. (2015). *Topcon products*. Recuperado el Julio de 2015, de <http://ag.topconpositioning.com/ag-products/guidance-systems>
- Topcon System 350*. (Julio de 2015). Obtenido de Topcon Precision Agriculture: <http://ag.topconpositioning.com/ag-products/guidance-systems/system-350>
- Torres-Sánchez, J., López-Granados, F., De Castro, A. I., & Peña-Barragán, J. M. (2013). Configuration and specifications of an unmanned aerial vehicle (UAV) for early site specific weed management. *PLoS ONE* , 8 (3).

- Trimble GNSS Systems*. (Julio de 2015). Obtenido de Trimble Website: <http://www.trimble.com/Agriculture/gnss-systems.aspx>
- Tsubota, R., Noguchi, N., & Mizushima, A. (2004). Automatic guidance with a laser scanner for a robot tractor in an orchard. *Proceedings of the Conference on Automation Technology for Off-Road Equipment*. Kyoto, Japan.
- Valente, J., Del Cerro, J., Barrientos, A., & Sanz, D. (2013). Aerial coverage optimization in precision agriculture management: A musical harmony inspired approach. *Computers and Electronics in Agriculture* , 99, 153-159.
- Vieri, M., Lisci, R., Rimediotti, M., & Sarri, D. (2012). The innovative RHEA airblast sprayer for tree crop treatment. *First International Conference on Robotic and Associated High-Technologies and Equipment for Agriculture*, (págs. 19-21). Pisa.
- Vieri, M., Lisci, R., Rimediotti, M., & Sarri, D. (2013). The RHEA-project robot for tree crops pesticide application. *Journal of Agricultural Engineering* , 44 (s1), 359-362.
- Wang, C., Soh, Y. C., Wang, H., & Wang, H. (2002). A hierarchical genetic algorithm for path planning in a static environment with obstacles. *Canadian Conference on IEEE CCECE 2002*. 3, págs. 1652-1657. IEEE.
- Washington, R., Golden, K., Bresina, J., Smith, D. E., Anderson, C., & Smith, T. (1999). Autonomous rovers for Mars exploration. *Proceedings of IEEE Aerospace Conference*, 1, págs. 237-251. Snowmass at Aspen, CO.
- Weiss, U., & Biber, P. (2011). Plant detection and mapping for agricultural robots using a 3D LIDAR sensor. *Robotics and autonomous systems* , 59 (5), 265-273.
- Weisz, R., Fleischer, S., & Smilowitz, Z. (1996). Site-specific integrated pest management for high-value crops: impact on potato pest management. *Journal of Economic Entomology* , 89 (2), 501-509.
- Wilkins, D. E., Lee, T. J., & Berry, P. (2003). Interactive Execution Monitoring of Agent Teams. *Journal of Artificial Intelligence Research* , 18, 217-261.
- Williams, B. C., & Nayak, P. P. (1996). A model-based approach to reactive self-configuring systems. *Proceedings of the National Conference on Artificial Intelligence*, (págs. 971-978).